NEW AGE

Ajit K. Mandal

INTRODUCTION TO CONTROL ENGINEERING

Modeling, Analysis and Design

NEW AGE INTERNATIONAL PUBLISHERS

INTRODUCTION TO

CONTROL ENGINEERING

INTRODUCTION TO

CONTROL ENGINEERING

Modeling, Analysis and Design

Ajit K. Mandal
Professor
Department of Electronics and Telecommunication Engineering
Faculty of Engineering and Technology
Jadavpur University
Kolkata, India

PUBLISHING FOR ONE WORLD

NEW AGE INTERNATIONAL (P) LIMITED, PUBLISHERS

New Delhi • Bangalore • Chennai • Cochin • Guwahati • Hyderabad
Jalandhar • Kolkata • Lucknow • Mumbai • Ranchi

Visit us at **www.newagepublishers.com**

Dedicated to:

the memory of my parents

K. L. Mandal

and

Rohini Mandal

in grateful reverence and appreciation

Preface

Control engineering is a very important subject to warrant its inclusion as a core course in the engineering program of studies in universities throughout the world. The subject is multidisciplinary in nature since it deals with dynamic systems drawn from the disciplines of electrical, electronics, chemical, mechanical, aerospace and instrumentation engineering. The common binding thread among all these divergent disciplines is a mathematical model in the form of differential or difference equations or linguistic models. Once a model is prepared to describe the dynamics of the system, there is little to distinguish one from the other and the analysis depends solely on characteristics like linearity or nonlinearity, stationary or time varying, statistical or deterministic nature of the system. The subject has a strong mathematical foundation and mathematics being a universal language; it can deal with the subject of interdisciplinary nature in a unified manner.

Even though the subject has strong mathematical foundation, emphasis throughout the text is not on mathematical rigour or formal derivation (unless they contribute to understanding the concept), but instead, on the methods of application associated with the analysis and design of feedback system. The text is written from the engineer's point of view to explain the basic concepts involved in feedback control theory. The material in the text has been organized for gradual and sequential development of control theory starting with a statement of the task of a control engineer at the very outset.

The book is intended for an introductory undergraduate course in control systems for engineering students. The numerous problems and examples have been drawn from the disciplines of electrical, electronics, chemical, mechanical, and aerospace engineering. This will help students of one discipline with the opportunity to see beyond their own field of study and thereby broaden their perceptual horizon. This will enable them to appreciate the applicability of control system theory to many facets of life like the biological, economic, and ecological control systems.

This text presents a comprehensive analysis and design of continuous-time control systems and includes more than introductory material for discrete systems with adequate guidelines to extend the results derived in connection with continuous-time systems. The prerequisite for the reader is some elementary knowledge of differential equations, vector-matrix analysis and mechanics.

Numerous solved problems are provided throughout the book. Each chapter is followed by review problems with adequate hints to test the reader's ability to apply the theory involved. Transfer function and state variable models of typical components and subsystems have been derived in the Appendix at the end of the book.

Most of the materials including solved and unsolved problems presented in the book have been class-tested in senior undergraduates and first year graduate level courses in the

field of control systems at the Electronics and Telecommunication Engineering Department, Jadavpur University.

The use of computer-aided design (CAD) tool is universal for practicing engineers and MATLAB is the most widely used CAD software package in universities through out the world. MATLAB scripts are provided so that students can learn to use it for calculations and analysis of control systems. Some representative MATLAB scripts used for solving problems are included at the end of each chapter whenever thought relevant. However, the student is encouraged to compute simple answers by hand in order to judge that the computer's output is sound and not 'garbage'. Most of the graphical figures were generated using MATLAB and some representative scripts for those are also included in the book. We hope that this text will give the students a broader understanding of control system design and analysis and prepare them for an advanced course in control engineering.

In writing the book, attempt has been made to make most of the chapters self-contained. In the introductory chapter, we endeavored to present a glimpse of the typical applications of control systems that are very commonly used in industrial, domestic and military appliances. This is followed by an outline of the task that a control-engineering student is supposed to perform. We have reviewed, in the second chapter, the common mathematical tools used for analysis and design of linear control systems. This is followed by the procedure for handling the block diagrams and signal flow graphs containing the transfer functions of various components constituting the overall system. In chapter 3, the concept of state variable representation along with the solution of state equations is discussed. The concept of controllability and observability are also introduced in this chapter along with the derivation of transfer function from state variable representation. The specifications for transient state and steady state response of linear systems have been discussed in chapter 4 along with the Bode technique for frequency domain response of linear control systems. In chapter 5, the concept of stability has been introduced and Routh-Hurwitz technique along with the Direct method of Lyapunov have been presented. Frequency domain stability test by Nyquist criteria has been presented in chapter 6. The root locus technique for continuous system has been discussed in chapter 7 and its extension to discrete cases has been included. The design of compensators has been taken up in chapter 8. In chapter 9, we present the concept of pole assignment design along with the state estimation. In chapter 10, we consider the representation of digital control system and its solution. In chapter 11, we present introductory material for optimal problem and present the solution of linear regulator problem. Chapter12 introduces the concepts of fuzzy set and fuzzy logic needed to understand Fuzzy Logic Control Systems presented in chapter 13.

The reader must be familiar with the basic tools available for analyzing systems that incorporate unwanted nonlinear components or deliberately introduced (relay) to improve system performance. Chapter 14 has been included to deal with nonlinear components and their analysis using MATLAB and SIMULINK through user defined s-functions. Finally, Chapter 15 is concerned with the implementation of digital controllers on finite bit computer, which will bring out the problems associated with digital controllers. We have used MATLAB and SIMULINK tools for getting the solution of system dynamics and for rapid verification of controller designs. Some notes for using MATLAB script M-files and function M-files are included at the end of the book.

The author is deeply indebted to a number of individuals who assisted in the preparation of the manuscript, although it is difficult to name everyone in this Preface. I would like to thank Saptarshi for his support and enthusiasm in seeing the text completed, Maya for the many hours she spent reviewing and editing the text and proof reading. I would like to thank many

people who have provided valuable support for this book project : Ms Basabi Banerjee for her effort in writing equations in MSword in the initial draft of the manuscript, Mr. U. Nath for typing major part of the manuscript and Mr. S. Seal for drawing some figures.

The author would like to express his appreciation to the former graduate students who have solved many problems used in the book, with special appreciation to Ms Sumitra Mukhopadhyay, who provided feedback and offered helpful comments when reading a draft version of the manuscript.

A. K. Mandal

Contents

CHAPTER

1

Control Systems and the Task of a Control Engineer

1.0 INTRODUCTION TO CONTROL ENGINEERING

The subject of control engineering is interdisciplinary in nature. It embraces all the disciplines of engineering including Electronics, Computer Science, Electrical Engineering, Mechanical Engineering, Instrumentation Engineering, and Chemical Engineering or any amalgamation of these. If we are interested to control the position of a mechanical load automatically, we may use an electrical motor to drive the load and a gearbox to connect the load to the motor shaft and an electronic amplifier to amplify the control signal. So we have to draw upon our working experience of electronic amplifier, the electrical motor along with the knowledge of mechanical engineering as to how the motor can be connected with the help of a gearbox including the selection of the gear ratio. If we are interested to regulate the DC output voltage of a rectifier to be used for a computer system, then the entire control system consists purely of electrical and electronics components. There will be no moving parts and consequently the response of such systems to any deviations from the set value will be very fast compared to the response of an electromechanical system like a motor. There are situations where we have to control the position of mechanical load that demands a very fast response, as in the case of aircraft control system. We shall recommend a hydraulic motor in place of an electrical motor for fast response, since the hydraulic motor has a bandwidth of the order of 70 radians/sec.

It is to be pointed out that the use of amplifiers is not the exclusive preserve of electronic engineers. Its use is widespread only because of the tremendous development in the discipline of electronics engineering over the last 30 to 40 years. Amplifiers may be built utilizing the properties of fluids resulting in hydraulic and pneumatic amplifiers. In petrochemical industry pneumatic amplifiers are a common choice while hydraulic amplifiers are widely used in aircraft control systems and steel rolling mills where very large torque are needed to control the position of a mechanical load. But whenever the measurement of any physical parameter of our interest is involved it should be converted to electrical signal at the first opportunity for subsequent amplification and processing by electronic devices. The unprecedented development of electronic devices in the form of integrated circuits and computers over the last few decades coupled with the tremendous progress made in signal processing techniques has made it extremely profitable to convert information about any physical parameter of our interest to electrical form for necessary preprocessing.

Since we are interested to control a physical parameter of our interest like temperature, pressure, voltage, frequency, position, velocity, concentration, flow, pressure, we must have suitable transducers to measure the variables and use them as suitable feedback signal for

proper control. Therefore, the knowledge of various transducers is essential to appreciate the intricacies of a practical control system.

In this text, we shall endeavor to introduce the basic principles of control systems starting from building mathematical models of elementary control systems and gradually working out the control strategy in a practical control system. We have drawn examples from various disciplines of engineering so as to put students from various disciplines of engineering on a familiar footing. We have assumed an elementary knowledge of differential calculus and the working knowledge for the solution of differential equations and elementary algebra.

The Control systems may be used in open loop or in close loop configuration. We shall explain these concepts by considering a schematic representation of a system, as shown in Fig. 1.1 that maintains the liquid level in a tank by controlling the incoming flow rate of fluid. But, before we explain its operation we shall highlight the importance of the concept of feedback first.

1.1 THE CONCEPT OF FEEDBACK AND CLOSED LOOP CONTROL

The concept of feedback is the single most important idea that governs the life of man in modern societies. In its present state of sophistication, human life would have been miserable without machines and most of the machines used by man could not be made to function with reasonable reliability and accuracy without the utilization of feedback. Most of the machines meant for domestic and industrial applications, for entertainment, health-services and military science, incorporate the concept of feedback. This concept is not exploited solely by man, it is also prevalent in nature and man has learnt it, like many other things, from nature. Our very life, for instance, is dependent on the utilization of feedback by nature.

Control systems may be classified as self-correcting type and non self-correcting type. The term self-correcting, as used here, refers to the ability of a system to monitor or measure a variable of interest and correct it automatically without the intervention of a human whenever the variable is outside acceptable limits. Systems that can perform such self-correcting action are called feedback systems or closed-loop systems whereas non self-correcting type is referred to as open loop system.

When the variable that is being monitored and corrected is an object's physical positions and the system involves mechanical movement is assigned a special name: *a servo system*.

1.2 OPEN-LOOP VERSUS CLOSED-LOOP SYSTEMS

Let us illustrate the essential difference between an open-loop system and a closed-loop system. Consider a simple system for maintaining the liquid level in a tank to a constant value by controlling the incoming flow rate as in Fig. 1.1(*a*). Liquid enters the tank at the top and flows out via the exit pipe at the bottom.

One way to attempt to maintain the proper level in the tank is to employ a human operator to adjust the manual valve so that the rate of liquid flow into the tank exactly balances the rate of liquid flow out of the tank when the liquid is at the desired level. It might require a bit of trial and error for the correct valve setting, but eventually an intelligent operator can set the proper valve opening. If the operator stands and watches the system for a while and observes that the liquid level stays constant, s/he may conclude that the proper valve opening has been set to maintain the correct level.

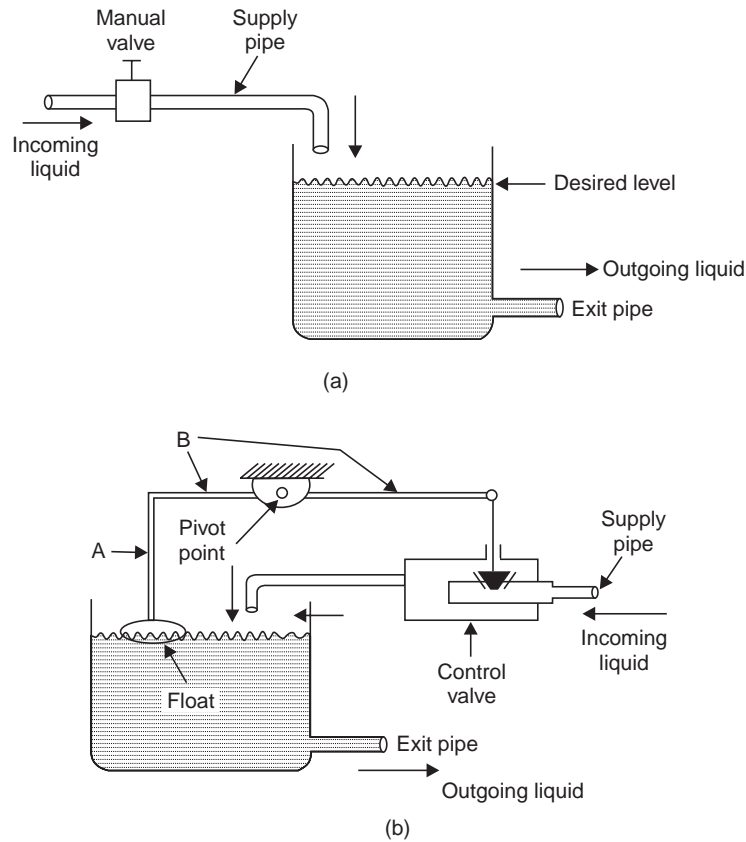



Fig. 1.1 System for maintaining the proper liquid level in a tank
(a) an open-loop system; it has no feedback and is not self-correcting.
(b) A closed-loop system; it has feedback and is self-correcting

In reality, however, there are numerous subtle changes that could occur to upset the balance s/he has taken trouble to achieve. For example, the supply pressure on the upstream side of the manual valve might increase for some reason. This would increase the input flow rate with no corresponding increase in output flow rate. The liquid level would start to rise and the tank would soon overflow. Of course, there would be some increase in output flow rate because of the increased pressure at the bottom of the tank when the level rises, but it would be a chance in a million that this would exactly balance the new input flow rate. An increase in supply pressure is just one example of a disturbing force that would upset the liquid level in the tank. There may be other disturbing forces that can upset the constant level. For instance, any temperature change would change the fluid viscosity and thereby changing the flow rates or a change in a system restriction downstream of the exit pipe would also change the output flow rate.

Now consider the setup in Fig. 1.1(*b*). If the liquid level falls a little too low, the float moves down, thereby opening the tapered valve to increase the inflow of liquid. If the liquid level rises a little too high, the float moves up, and the tapered valve closes a little to reduce the inflow of liquid. By proper construction and sizing of the valve and the mechanical linkage between float and valve, it would be possible to control the liquid level very close to the desired set point. In this system the operating conditions may change causing the liquid level to deviate from the desired point in either direction but the system will tend to restore it to the set value.

Our discussion to this point has been with respect to the specific problem of controlling the liquid level in a tank. However, in general, many different industrial control systems have certain things in common. Irrespective of the exact nature of any control system, there are certain relationships between the controlling mechanisms and the controlled variable that are similar. We try to illustrate these cause-effect relationships by drawing block diagrams of our industrial systems. Because of the "similarity" among different systems, we are able to devise generalized block diagrams that apply to all systems. Such a generalized block diagram of an open loop system is shown in Fig. 1.2(*a*).

The block diagram is basically a cause and effect indicator, but it shows rather clearly that for a given setting the value of the controlled variable cannot be reliably known in presence of disturbances. Disturbances that happen to the process make their effects felt in the output of the process—the controlled variable. Because the block diagram of Fig. 1.2(*a*) does not show any lines coming back around to make a circular path, or to "close the loop," such a system is called an open-loop system. All open-loop systems are characterized by its inability to compare the actual value of the controlled variable to the desired value and to take action based on that comparison. On the other hand, the system containing the float and tapered valve of Fig. 1.1(*b*) is capable of this comparison. The block diagram of the system of Fig. 1.1(*b*) is shown in Fig. 1.2(*b*). It is found from the diagram that the setting and the value of the controlled variable are compared to each other in a comparator. The output of the comparator represents the difference between the two values. The difference signal, called "actuating signal", then feeds into the controller allowing the controller to affect the process.

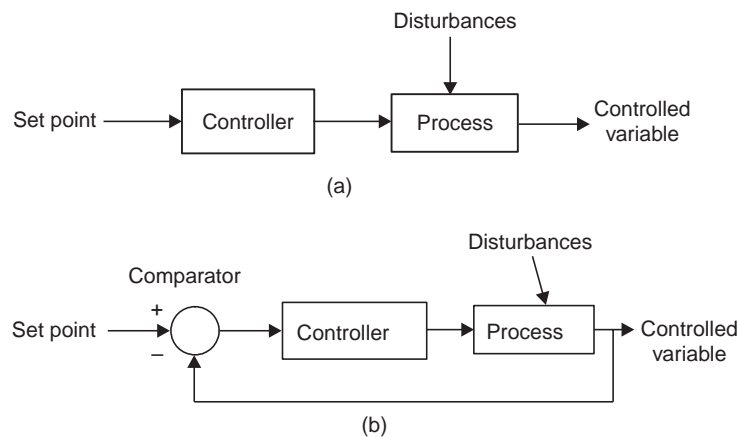



Fig. 1.2 Block diagrams that show the cause-effect relationships between the different parts of the system (a) for an open-loop system (b) for a closed-loop system

The fact that the controlled variable comes back around to be compared with the setting makes the block diagram look like a "closed loop". A system that has this feature is called a closed-loop system. All closed-loop systems are characterized by the ability to compare the actual value of the controlled variable to its desired value and automatically take action based on that comparison. The comparator performs the mathematical operation of summation of two or more signals and is represented by a circle with appropriate signs.

For our example of liquid level control in Fig. 1.1(*b*), the setting represents the location of the float in the tank. That is, the human operator selects the level that s/he desires by locating the float at a certain height above the bottom of the tank. This setting could be altered

by changing the length of rod A that connects the float to horizontal member B of the linkage in Fig. 1.1(*b*).

The comparator in the block diagram is the float itself together with the linkages A and B in our example. The float is constantly monitoring the actual liquid level, because it moves up or down according to that level. It is also comparing with the setting, which is the desired liquid level, as explained above. If the liquid level and setting are not in agreement, the float sends out a signal that depends on the magnitude and the polarity of the difference between them.

That is, if the level is too low, the float causes horizontal member B in Fig. 1.1(*b*) to be rotated counterclockwise; the amount of counterclockwise displacement of B depends on how low the liquid is. If the liquid level is too high, the float causes member B to be displaced clockwise. Again, the amount of displacement depends on the difference between the setting and the controlled variable; in this case the difference means how much higher the liquid is than the desired level.

Thus the float in the mechanical drawing corresponds to the comparator block in the block diagram of Fig. 1.2(*b*). The controller in the block diagram is the tapered valve in the actual mechanical drawing.

In our particular example, there is a fairly clear correspondence between the physical parts of the actual system and the blocks in the block diagram. In some systems, the correspondence is not so clear-cut. It may be difficult or impossible to say exactly which physical parts comprise which blocks. One physical part may perform the function of two different blocks, or it may perform the function of one block and a portion of the function of another block. Because of the difficulty in stating an exact correspondence between the two system representations, we will not always attempt it for every system we study.

The main point to be realized here is that when the block diagram shows the value of the controlled variable being fed back and compared to the setting, the system is called a closed-loop system. As stated before, such systems have the ability to automatically take action to correct any difference between actual value and desired value, no matter why the difference occurred.

Based on this discussion, we can now formally define the concept of feedback control as follows:

Definition 1 *The feedback control is an operation, which, in the presence of disturbing forces, tends to reduce the difference between the actual state of a system and an arbitrarily varied desired state of the system and which does so on the basis of this difference.*

In a particular system, the desired state may be constant or varying and the disturbing forces may be less prominent. A control system in which the desired state (consequently the set point) is constant, it is referred to as a **regulator** (example: a regulated power supply) and it is called a tracking system if the set point is continuously varying and the output is required to track the set point (example: RADAR antenna tracking an aircraft position).

Figure 1.3 shows another industrial process control system for controlling the temperature of a pre-heated process fluid in a jacketed kettle. The temperature of the process fluid in the kettle is sensed by transducers like a thermocouple immersed in the process fluid. Thermocouple voltage, which is in tens of milli-volts, represents the fluid temperature and is amplified by an electronic DC amplifier A_{fb} to produce a voltage V_b. The battery and the potentiometer provide a reference (set point) voltage V_r, which is calibrated in appropriate temperature scale. The input voltage V_e to the amplifier A_e is the difference of the reference voltage and the feedback voltage V_b. The output voltage V_o of this amplifier is connected to the

solenoid coil that produces a force, f_s, proportional to the current through the coil i_a. The solenoid pulls the valve plug and the valve plug travels a distance **x**. The steam flow rate **q** through the valve is directly proportional with the valve-opening **x**. The temperature θ of the process fluid will be proportional to the valve opening.

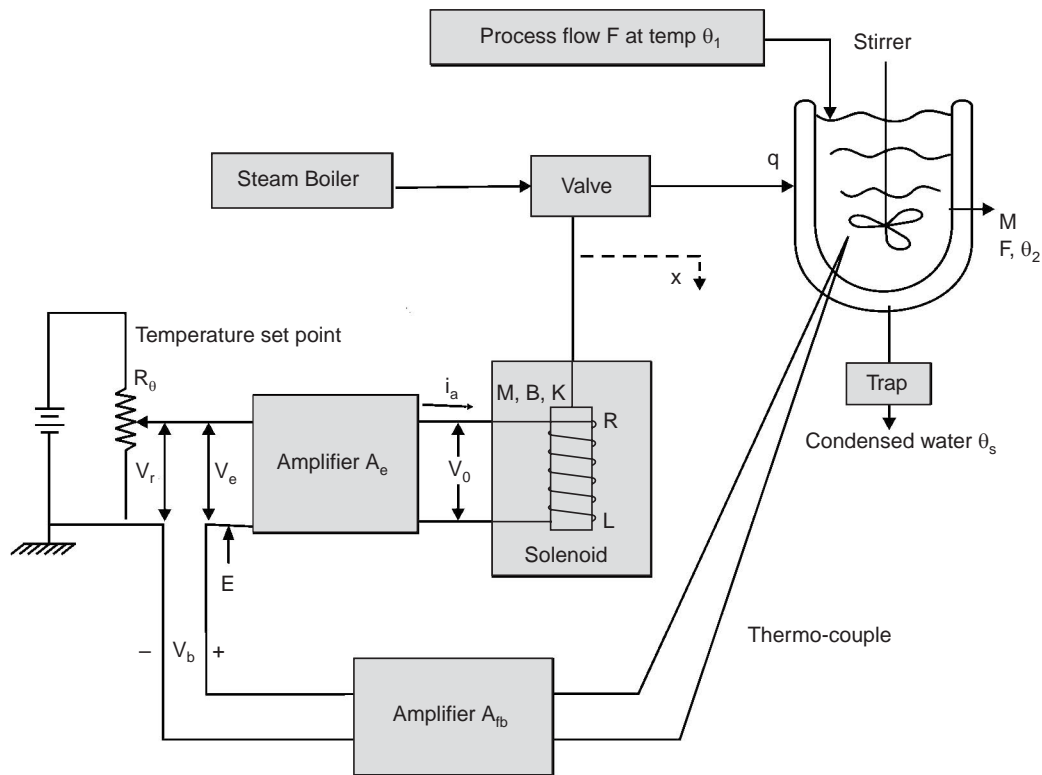



Fig. 1.3 Feedback control system for temperature control of a process fluid

If the flow rate, F of the process fluid increases, the temperature of the kettle will decrease since the steam flow has not increased. The action of the control system is governed in such a way as to increase the steam flow rate to the jacketed kettle, until temperature of the process fluid is equal to the desired value set by the reference potentiometer. Let us see the consequences to the increased flow rate F of the process fluid. As the temperature in the kettle falls below the set point, the thermocouple output voltage and its amplified value V_b will decrease in magnitude. Since $V_e = V_r - V_b$ and V_r is fixed in this case by the set point potentiometer, a decrease in V_b causes V_e to increase. Consequently the amplifier output voltage V_o increases, thereby increasing the travel **x** of the valve plug. As a result, the opening of the valve increases, resulting in an increase to the steam flow rate **q,** which increases the temperature of the process fluid.

If the flow rate of the process fluid decreases and the temperature inside the kettle increases, V_b increases and V_e decreases. The output of the second amplifier V_o and the travel x of the valve plug decreases with a consequent reduction of the valve opening. The steam flow-rate q, therefore, decreases and results in a decrease of the kettle temperature until it equals to the set temperature.

The open loop mode of temperature control for the process fluid is possible by removing the sensing thermocouple along with the associated amplifier A_{fb} from the system. The point E of the amplifier A_e is returned to the ground and the potentiometer is set to the desired temperature. Since in this case V_e is equal to a constant value V_r set by the potentiometer, the output amplifier voltage V_o and travel x of the valve plug is fixed making the opening of the valve fixed. The steam flow rate q is also fixed and the temperature of the kettle will be fixed if the process fluid flow-rate F is maintained to the constant value. This control arrangement is adequate if the flow rate of the process fluid is maintained to constant value. In actual practice, however, the flow rate deviates from a constant value and the temperature fluctuates with the fluctuation in F. The temperature will, in general, differ from the set point and an experienced human operator will have to change the set point interactively. When the disturbances to the system are absent, the final temperature of the kettle will be determined by the experienced skill of the operator and accuracy of calibration of the temperature-setting potentiometer R_θ . But in presence of disturbances, which are always there, no amount of skill and experience of the operator will be adequate and the process product is likely to be adversely affected.

The closed loop control system in Fig. 1.3 may be represented by block diagrams as shown in Fig. 1.4. The desired value of the system state is converted to the reference input by the transducers known as reference input elements.

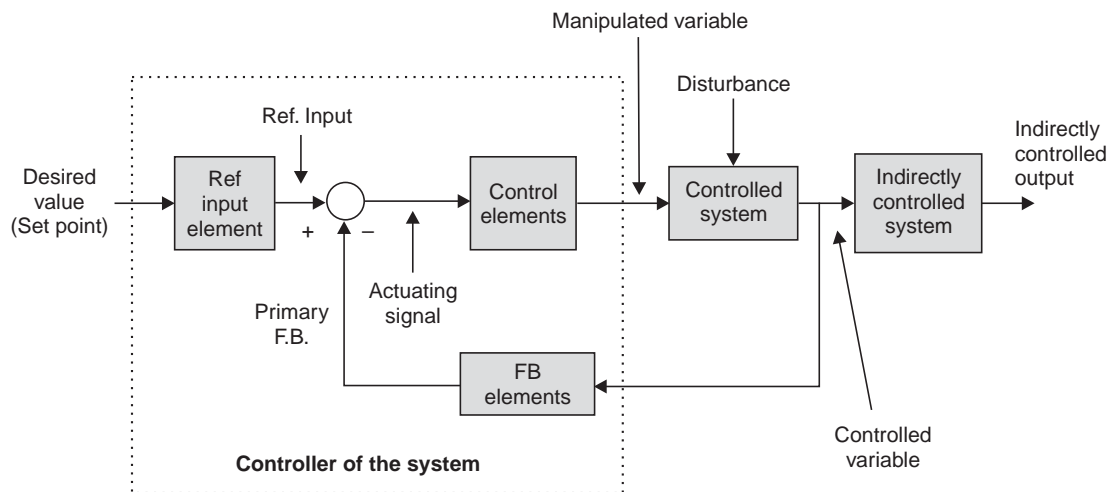



Fig. 1.4 Block diagram of a closed loop system

The controlled variable is converted to the feedback signal by the feedback element, also it is converted to the indirectly controlled variable, which is actual output of the entire feedback control system. The subtraction of feedback signal and reference input to obtain the actuating signal is indicated by a small circle with a sign to represent the arithmetic operation. The parts of the diagram enclosed by the dotted line constitute the controller of the system.

1.3 FEEDFORWARD CONTROL

With reference to the temperature control system in Fig. 1.3, the effect of the disturbance in flow rate F is manifested by the change in temperature of the process fluid for a constant value

of steam input. But the effect of this disturbance cannot be immediately sensed in the output temperature change due to the large time constant in the thermal process. Besides, the effect of corrective action in the form of change of steam input will be felt at the processor output at a time when the temperature might have deviated from the set point by a large value. The disturbance in the flow rate depends on the level of the fluid in the tower (Fig. 1.5) and can be easily measured by a flow meter. If we generate a corrective signal by an open loop controller block with a very small time constant and use it as input to the temperature controller ($G_f(s)$ in Fig. 1.5(*b*)), the transient response of the process temperature might be controlled within a tighter limit. This type of control is known as feedforward control. The motivation behind the feedforward control is to provide the corrective action for the disturbance, if it is measurable, not by using the delayed feedback signal at the output of the process but by using some other controller block with fast response. This strategy of using a faster control path will provide a better transient response in the system output. As soon as a change in the flow rate in the input fluid occurs, corrective action will be taken simultaneously, by adjusting the steam input to the heat exchanger. This can be done by feeding both the signal from the flow meter and the signal from the thermocouple to the temperature controller.

The feedforward controller block $G_f(s)$ in part (*b*) of the Fig. 1.5, is found to be in the forward path from the disturbance input to the process output.

Feedforward control can minimize the transient error, but since feedforward control is open loop control, there are limitations of its functional accuracy. Feedforward control will not cancel the effects of unmeasurable disturbances under normal operating condition. It is, therefore, necessary that a feedforward control system include a feedback loop as shown in Fig. 1.5. The feedback control compensates for any imperfections in the functioning of the open loop feedforward control and takes care of any unmeasurable disturbances.

Fig. 1.5 (*Contd.*)

(b)

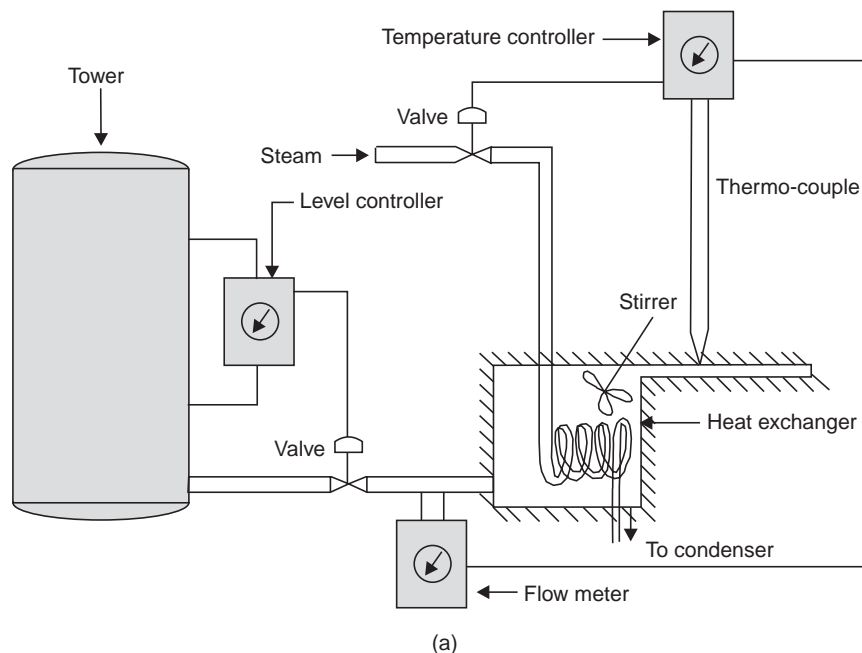
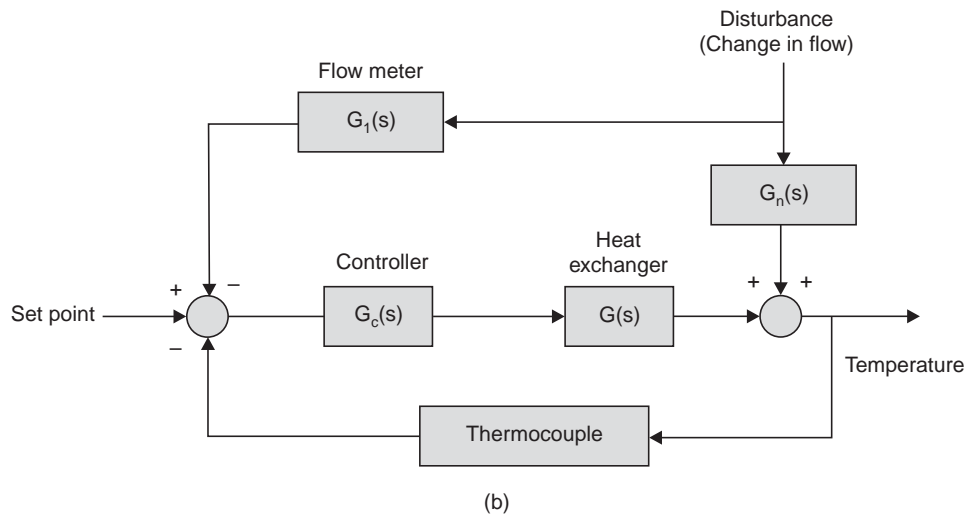



Fig. 1.5 (a) Feedforward temperature control system (b) Block diagram

1.4 FEEDBACK CONTROL IN NATURE

The examples of feedback control are abundant in nature and it plays a very important role for controlling the activities of animals including man. The life of human being itself is sustained by the action of feedback control in several forms. To cite an example, let us consider the maintenance of body temperature of warm-blooded animals to a constant value. The body temperature of human being is maintained around 38.5°C in spite of the wide variation of ambient temperature from minus 20°C to plus 50°C.

The feedback control system is absent in the cold-blooded animals like lizards, so their body temperature varies with the ambient temperature. This severely restricts their activities; they go for hibernation in the winter when their body processes slow up to the extreme end. They become active again in the summer, and the warmer part of each day.

The body temperature of a human being is regulated to a constant value by achieving a balance in the process of generation and dissipation of heat. The primary source of heat in the case of living creatures is the metabolic activity supplemented by muscle activity and the heat is dissipated from the body by radiation, conduction and convection.

When the ambient temperature suddenly rises, the skin senses this increase and sets in a series of operation for dissipating body-heat. Thermal radiation takes place directly from the body surface to the surrounding. The body fluid and the blood flow take part in the convection process. The blood vessels are constricted and the flow is diverted towards the outer surface of the body. The metabolic activity decreases, decreasing the heat generation. The respiration rate increases, so that more heat can be dissipated to the mass of air coming in contact with the lung surface, Blood volume is increased with body fluid drawn into circulation resulting in further cooling. The perspiration rate increases taking latent heat from the body surface thereby decreasing the body temperature. In a similar way when the outside temperature falls, extra heat is chemically generated by increased rate of metabolism. The heat generation is also supplemented by shivering and chattering of the teeth.

Other examples of feedback control in human body include: Hydrogen-ion concentration in blood, concentration of sugar, fat, calcium, protein and salt in blood. Some of these variables should be closely controlled and if these were not so convulsions or coma and death would

result. Man's utilization of the feedback control is not as critical as in nature, except probably, in the space and undersea voyages. But utilization of feedback control is gaining importance with the increased sophistication of modern life and soon the role of feedback control in shaping the future of man will be as critical as is found in nature.

1.5 A GLIMPSE OF THE AREAS WHERE FEEDBACK CONTROL SYSTEMS HAVE BEEN EMPLOYED BY MAN

The following list gives a glimpse of the areas of human activities where the feedback control system is extensively used.

Domestic applications:

Regulated voltage and frequency of electric power, thermostat control of refrigerators and electric iron, temperature and pressure control of hot water supply in cold countries, pressure of fuel gas, automatic volume and frequency control of television, camcorder and radio receivers, automatic focusing of digital cameras.

Transportation:

Speed control of the airplane engines with governors, control of engine pressure, instruments in the pilot's cabin contain feedback loops, control of rudder and aileron, engine cowl flaps, instrument-landing system.

In sea going vessels:

Automatic steering devices, radar control, Hull control, boiler pressure and turbine speed control, voltage control of its generators.

In automobiles:

Thermostatic cooling system, steering mechanisms, the gasoline gauge, and collision avoidance, idle speed control, antiskid braking in the latest models and other instruments have feedback loops.

Scientific applications:

Measuring instruments, analog computers, electron microscope, cyclotron, x-ray machine, x-y plotters, space ships, moon-landing systems, remote tracking of satellites.

In industry:

Process regulators, process and oven regulators, steam and air pressure regulators, gasoline and steam engine governors, motor speed regulators, automatic machine tools such as contour followers, the regulation of quantity, flow, liquid level, chemical concentration, light intensity, colour, electric voltage and current, recording or controlling almost any measurable quantity with suitable transducers.

Military applications:

Positioning of guns from 30 caliber machine guns in aircraft to mighty 16-inch guns abroad battle ships, search lights, rockets, torpedoes, surface to air missiles, ground to air or air to air missiles, gun computers, and bombsights, and guided missiles.

1.6 CLASSIFICATION OF SYSTEMS

For convenience of description and mathematical analysis, systems are classified into different categories. They are *classified* according to the *nature of inputs, number of inputs, number of outputs* and some *inherent characteristic* of the system. Fig. 1.6 shows the block diagram

representation of a system with only a single input $u(t)$ and a single output $y(t)$ (SISO). Similarly, we might have multi inputs and multi outputs (MIMO) systems, single input multi output (SIMO) and multi input and single output (MISO) systems.

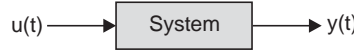



Fig. 1.6 Block diagram representation of a system

In the SISO system in Fig. 1.6, it is assumed that it has no internal sources of energy and is at rest, prior to the instant at which the input $u(t)$ is applied. The cause and effect relationship may be represented in short as

$$y(t) = \text{L}u\ (t) \tag{1.1}$$

where L is an *operator* that characterizes the system. It may be a function of u, y or time t and may include operations like differentiation and integration or may be given in probabilistic language.

A system is *deterministic* if the output $y(t)$ is unique for a given input $u(t)$. For *probabilistic* or *non-deterministic* system, the output $y(t)$ is not unique but probabilistic, with a probability of occurrence for a given input. If the input to a deterministic system is statistical, like, noise, the output is also statistical in nature.

A system is said to be *non-anticipative* if the present output is not dependent on future inputs. That is, the output $y(t_o)$ at any instant t_o is solely determined by the characteristics of the system and the input $u(t)$ for $t > t_o$. In particular, if $u(t) = 0$ for $t > t_o$; then $y(t) = 0$.

An *anticipatory* system cannot be realized since it violates the normal cause and effect relationship.

1.6.1 Linear System

Let us assume that the outputs of the systems in Fig. 1.6 are $y_1(t)$ and $y_2(t)$ respectively corresponding to the inputs $u_1(t)$ and $u_2(t)$. Let k_1 and k_2 represent two arbitrary constants. The system in Fig. 1.6 will be linear if the output response of the system satisfies the *principle of homogeneity*

$$\text{L}(k_1 u_1(t)) = k_1 \text{L}(u_1(t)) \tag{1.2a}$$

and the *principle of additivity*

$$\text{L}[k_1 u_1(t) + k_2 u_2(t)] = k_1 \text{L}[u_1(t)] + k_2 \text{L}[u_2(t)] \tag{1.2b}$$

The principle in (1.2*b*) is also known as the *principle of superposition*.

In other words, a system is linear if its response $y(t)$ is multiplied by k_1 when its input is multiplied by k_1. Also the response follows the principle of superposition, that is, $y(t)$ is given by $k_1 y_1(t) + k_2 y_2(t)$ when the input $u(t)$ becomes $k_1 u_1(t) + k_2 u_2(t)$ for all u_1, u_2, k_1 and k_2.

If the principle of homogeneity together with the principle of superposition holds good for a certain range of inputs u_1 and u_2, the system is linear in that range of inputs and non-linear beyond that range.

1.6.2 Time-Invariant System

For a time-invariant or fixed system, the output is not dependent on the instant at which the input is applied. If the output at t is $y(t)$ corresponding to an input $u(t)$ then the output for a fixed system will be

$$\text{L}\ u(t - \lambda) = y(t - \lambda) \tag{1.3}$$

A system, which is not time-invariant, is a time-varying one. A few examples of the above definitions are considered below:

Example 1.1 A differentiator is characterized by

$$y(t) = \frac{d}{dt}\, u(t)$$

Here, the operator L = $\frac{d}{dt}$. Therefore, $y_1(t) = \frac{d}{dt}\,[k_1 u_1(t)] = k_1 \frac{d}{dt}\, u_1(t)$

and $$\frac{d}{dt}\,[k_1 u_1(t) + k_2 u_2(t)] = k_1 \frac{d}{dt}\, u_1(t) + k_2 \frac{d}{dt}\, u_2(t)$$

Hence the system is linear. It is also realizable and time-invariant.

Example 1.2 A system is characterized by $y(t) = [u(t)]^2$.

In this case, the operator L is the squarer and since $[k_1 u_1(t)]^2 \neq k_1[u_1(t)]^2$ and $[k_1 u_1(t) + k_2 u_2(t)]^2 \neq k_1 u_1^2(t) + k_2 u_2^2(t)$, the system is nonlinear.

It is realizable and time invariant.

Example 1.3 A system is characterized by the relationship

$$y(t) = t\,\frac{d}{dt}\, u(t)$$

It is linear since, $y_1(t) = t\,\frac{d}{dt}\,[k_1 u_1(t)] = tk_1 \frac{d}{dt}\,[u_1(t)] = k_1 t \frac{d}{dt}\,[u_1(t)]$

and $$t\,\frac{d}{dt}\,[k_1 u_1(t) + k_2 u_2(t)] = k_1 t \frac{d}{dt}\, u_1(t) + k_2 t \frac{d}{dt}\, u_2(t)$$

It is realizable but time varying since $t\,\frac{d}{dt}\, u(t - \lambda) \neq (t - \lambda)\,\frac{du(t - \lambda)}{d(t - \lambda)}$

Example 1.4 A system is characterized by the relationship

$$y(t) = u(t)\,\frac{d}{dt}\, u(t)$$

The system is nonlinear since $k_1 u_1(t)\,\frac{d}{dt}\, k_1 u_1(t) \neq k_1 u_1(t)\,\frac{d}{dt}\, u_1(t)$

and $$[k_1 u_1(t) + k_2 u_2(t)]\,\frac{d}{dt}\,[k_1 u_1(t) + k_2 u_2(t)] \neq k_1 u_1(t)\,\frac{d}{dt}\, u_1(t) + k_2 u_2(t)\,\frac{d}{dt}\, u_2(t)$$

It is realizable and time-invariant.

Systems are also classified based on the nature of signals present at all the different points of the systems. Accordingly systems may be described as continuous or discrete. A signal is continuous if it is a function of a continuous independent variable t [see Fig. 1.7(*a*)]. The above definition of continuous signal is broader than the mathematical definition of continuous function. For example the signal $f(t)$ in Fig. 1.7(*a*) is continuous in the interval $t_1 < t < t_2$, but it is not a continuous *function* of time in the same interval. A signal is discrete if it is a function of time at discrete intervals [see Fig. 1.7(*b*)].

A system is continuous if the signals at all the points of the system are a continuous function of time and it will be referred to as a discrete system if the signal at any point of the system is discrete function of time.

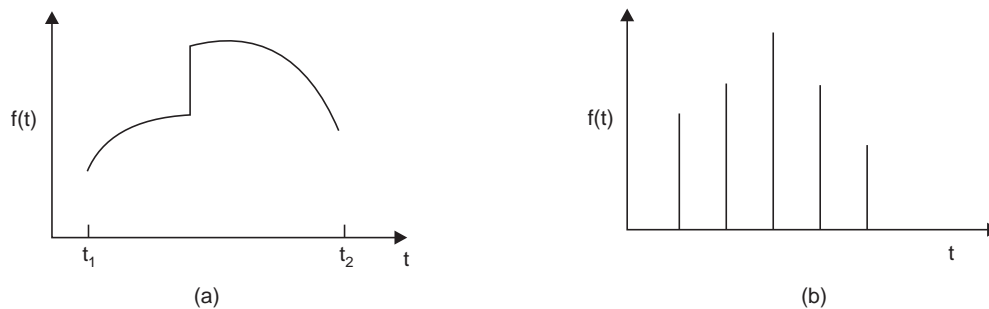



Fig. 1.7 Classification of systems based on nature of signals; (a) continuous signal and (b) discrete signal

1.7 TASK OF CONTROL ENGINEERS

In order to put the control engineers in proper perspective, we endeavor to present, at the very outset, the control-engineering problem together with the task to be performed by a control engineer.

(1) Objective of the Control System

A control system is required to do a job and is specified to the control engineer by the user of the control system. The engineer is expected to assemble the various components and sub systems, based on the laws of the physical world, to perform the task. The quality of performance is normally specified in terms of some mathematical specifications, that the control system is required to satisfy. The specifications may be the accuracy of controlled variable in the 'steady state' (the behavior of the system as time goes to infinity following a change in set point or disturbance) or it is concerned with the 'transient response'- the way the control variable is reaching the steady state following some disturbance or change in set value.

(2) Control Problem

Since the control system is required to satisfy a performance specification expressed in mathematical terms, the control engineer needs to solve a mathematical problem.

(3) System Modeling

Since some mathematical problems are to be solved, a mathematical model of the dynamics of the control system components and subsystems are to be formulated. The differential equation and the state variable representation are very popular mathematical models for describing the dynamics of a control system. The transfer function model is applicable if the system is linear. If the description of the system behavior is linguistic then fuzzy logic and fuzzy model of the system will be needed [1-3].

(4) System Analysis

Once the mathematical model of the basic system is obtained, its analysis is performed by using the existing mathematical tools available to the control engineer to study its behavior. The analysis may be carried out in the frequency domain or in time domain.

(5) Modification of the Control System

If the analysis reveals some sort of shortcoming in meeting one or more of the performance specifications in the steady state and /or transient state, the basic system needs to be modified by incorporating some additional feedback or by incorporating compensator to modify the system

behavior. The lag-lead compensator and state feedback design method are widely used for improving the system performance.

(6) Optimal Control

Among a number of alternative design solutions, some control laws are superior to others so far as the performance specifications are concerned. This leads to the problem of optimal control, where among all the possible solutions, the solution that optimizes the performance specification (like minimization of energy consumption, maximization of production rate or minimization of time and minimization of waste of material) should be chosen. Consider, for instance, the problem of landing of a space vehicle on the surface of the moon from earth. Since the consumption of fuel is an important consideration for space journey, of the innumerable trajectories from the earth to the surface of the moon, the one that will consume minimum fuel will be chosen and the controller will be programmed for that trajectory. In Haber's process of manufacturing Ammonia, the yield per unit time is dependent on the temperature of the reaction chamber. Control laws may be designed to maintain temperature of the reaction chamber such that the yield is maximum.

(7) Adaptive Control

In some, control systems the process parameters change with time or environmental conditions. The controller designed by assuming fixed system parameters fails to produce acceptable performance. The controller for such systems should adapt to the changes in the process such that the performance of the control system is not degraded in spite of the changes in the process. This gives rise to the problem of designing adaptive controller for a system. For example, the transfer functions of an aircraft changes with its level of flight and velocity, so that the effectiveness of the pilot's control stick will change with the flight conditions and the gain of the controller is to be adapted with the flight conditions [4]. In the space vehicle the fuel is an integral part of the mass of the vehicle, so the mass of the vehicle will change with time. An Adaptive controller that adapts itself to the changes in mass is expected to perform better. The design of adaptive controller, therefore, becomes an important issue in control engineering problem.

(8) Fuzzy Control

When the system description and performance specification are given in linguistic terms, the control of the system can better be handled by Fuzzy Logic setting down certain rules characterizing the system behavior and common sense logic (vide Chapter 12 and Chapter 13).

1.8 ALTERNATIVE WAYS TO ACCOMPLISH A CONTROL TASK

(*i*) **Process temperature control:** Let us now consider the problem of controlling the temperature of a process fluid in a jacketed kettle (Fig. 1.3). The problem can be solved in a number of alternative ways depending on the form of energy available to the control engineer.

(*a*) *The form of energy available is electricity:* If the available source of energy is electricity, the temperature of the fluid inside the kettle may be controlled by manipulating the current through a heater coil. The average power supplied to the heater coil can be manipulated by varying the firing angle of a Triac or Silicon Controlled Rectifier (SCR with a suitable triggering circuit).

(*b*) *The form of energy available is steam:* The temperature of the fluid inside the kettle will be proportional to the volume of steam coming into contact with the process fluid in the kettle. By connecting a solenoid-operated valve in the pipe through which steam flows, the volume of steam flow per unit time may be manipulated by opening and closing of the valve.

(*c*) *The form of energy available is fuel oil:* The temperature of an oil-fired furnace may be controlled by controlling the fuel to air ratio as well as the flow rate of fuel oil. By sensing the temperature of the chemical bath one can adjust the flow rate of fuel and its ratio to air by means of a solenoid operated valve to control the temperature of the furnace directly and the temperature of the fluid inside the kettle indirectly.

(*d*) *The solar energy as the power source:* The solar energy, concentrated by a set of focusing lens after collecting it from reflectors, may be used to control the temperature of the fluid inside the kettle. The amount of solar energy may be regulated by controlling the position of the reflectors as the position of the sun in the sky changes.

It is, therefore, apparent that the job of controlling the temperature of a fluid in a jacketed kettle may be accomplished in a number of alternative ways. Depending on the form of energy available, any of the above methods may be adopted for a given situation. The choice of a particular method depends on many other factors like technical feasibility and economic viability. The important point that is to be emphasized here is that the control objective may be realized by using a number of alternative methods. In each method, the components and subsystems should be assembled, in a meaningful way, by utilizing the knowledge of the physical world.

(*ii*) **Room temperature control:** Let us consider the problem of controlling room temperature using a room air conditioner as another example. The compressor of the room air conditioner may be switched off if the room temperature is less than the set temperature and switched on if the room temperature is higher than the set value. The compressor is kept on until the room temperature is equal to the set temperature. A thermostat switch senses the temperature in the room and when the temperature of the room goes below the set point, the power to the compressor is again switched off. The variation of ambient temperature outside the room and the escape of cool air due to opening of the door are the external disturbances to the control system and the change in the number of occupants in the room is the load disturbance. The temperature of the room could have been controlled by a central air conditioning system, where the volume of cool air entering the room could have been controlled by opening and closing of a solenoid operated valve.

1.9 A CLOSER LOOK TO THE CONTROL TASK

The presence of reference input element is always implied in a system and with the directly controlled output variable taken as the system output, the block diagram of Fig. 1.4 is simplified

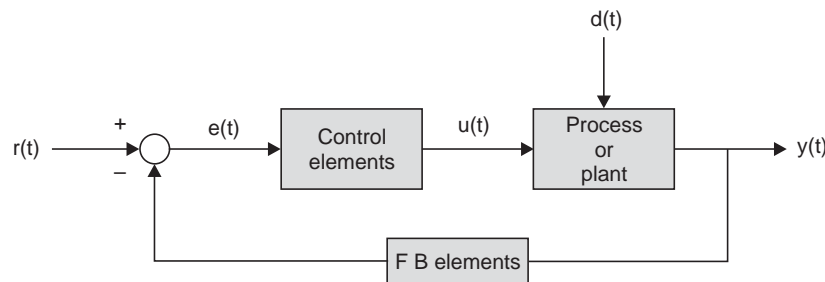



Fig. 1.8 Block diagram of a basic control system

and shown as the basic control system in Fig. 1.8. The process (or "plant") is the object to be controlled. Its inputs are $u(t)$, its outputs are $y(t)$, and reference input is $r(t)$. In the process fluid control problem, $u(t)$ is the steam input to the jacketed kettle, $y(t)$ is the temperature of

the process fluid and $r(t)$ is the desired temperature specified by the user. The plant is the jacketed kettle containing the process fluid. The controller is the thermocouple, amplifiers and solenoid valve (elements inside the dotted line in Fig. 1.4). In this section, we provide an overview of the steps involved in the design of the controller shown in Fig. 1.8. Basically, these are modeling, controller design, and performance evaluation.

1.9.1 Mathematical Modeling

After the control engineer has interconnected the components, subsystems and actuators in a meaningful way to perform the control job, often one of the next tasks that the designer undertakes is the development of a mathematical model of the process to be controlled, in order to gain understanding of the problem. There are only a few ways to actually generate the model. We can use the laws of the physical world to write down a model (*e.g.*, $F = mf$). Another way is to perform "system identification" via the use of real plant data to produce a model of the system [5]. Sometimes a combined approach is used where we use physical laws to write down a general differential equation that we believe represents the plant behavior, and then we perform experiments in the plant to determine certain model parameters or functions.

Often more than one mathematical model is produced. An actual model is one that is developed to be as accurate as possible so that it can be used in simulation-based evaluation of control systems. It must be understood, therefore, that there is never a perfect mathematical model for the plant. The mathematical model is abstraction and hence cannot perfectly represent all possible dynamics of any physical process (*e.g.*, certain noise characteristics or failure conditions).

Usually, control engineers keep in mind that they only need to use a model that is accurate enough to be able to design a controller that works[†]. However, they often need a very accurate model to test the controller in simulation before it is tested in an experimental setting. Lower-order design-models may be developed satisfying certain assumptions (*e.g.*, linearity or the inclusion of only certain forms of non-linearities) that will capture the essential plant behavior. Indeed, it is quite an art and science to produce good low-order models that satisfy these constraints. Linear models such as the one in Equation (1.4) has been used extensively in the modern control theory for linear systems and is quite mature [vide Chapter 3].

$$\begin{aligned} \dot{x}(t) &= A\,x(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \tag{1.4}$$

[†]In the classic optimization problem of "traveling sales representative", it is required to minimize the total distance traveled by considering various routes between a series of cities on a particular trip. For a small number of cities, the problem is a trivial exercise in enumerating all the possibilities and choosing the shortest route. But for 100 cities there are factorial 100 (or about 10^{200}) possible routes to consider! No computers exist today that can solve this problem through a brute-force enumeration of all the possible routes. However, an optimal solution with an accuracy within 1 percent of the exact solution will require two days of CPU time on a supercomputer which is about 40 times faster than a personal computer for finding the optimum path (*i.e.*, minimum travel time) between 100,00 nodes in a travel network. If the same problem is taken and the precision requirement is increased by a modest amount to value of 0.75 percent, the computing time approaches seven months! Now suppose we can live with an accuracy of 3.5 percent and increase the nodes in the network to 1000,000; the computing time for this problem is only slightly more than three hours [6]. This remarkable reduction in cost (translating time to money) is due solely to the acceptance of a lesser degree of precision in the optimum solution. The big question is can humans live with a little less precision? The answer to this question depends on the situation, but for the vast majority of problems we encounter daily, the answer is a resounding yes.

In this case u is the m-dimensional input; x is the n-dimensional state; y is the p-dimensional output; and A, B, C and D are matrices of appropriate dimension. Such models, or transfer functions ($G(s) = C(sI - A)^{-1} B + D$ where s is the Laplace variable), are appropriate for use with frequency domain design techniques (*e.g.*, Bode plot and Nyquist plots), the root-locus method, state-space methods, and so on. Sometimes it is assumed that the parameters of the linear model are constant but unknown, or can be perturbed from their nominal values, then techniques for "robust control" or adaptive control are developed [7-8].

Much of the current focus in control is on the development of controllers using nonlinear models of the plant of the form:

$$\dot{x}(t) = f\{x(t), u(t)\}$$
$$y(t) = g\{x(t), u(t)\} \quad (1.5)$$

where the variables are defined as in the linear model and f and g are nonlinear function of their arguments. One form of the nonlinear model that has received significant attention is

$$\dot{x}(t) = f(x(t)) + g(x(t))\, u(t) \quad (1.6)$$

since it is possible to exploit the structure of this model to construct nonlinear controllers (*e.g.*, in feedback linearization or nonlinear adaptive control). Of particular interest with both the nonlinear models above is the case where f and g are not completely known and subsequent research focuses on robust control of nonlinear systems.

Discrete time versions of the above models are also used when a digital computer is used as a controller and stochastic effects are often taken into account via the addition of a random input or other stochastic effects.

Stability is the single most important characteristic of a system and the engineer should pay attention to it at the very early stage of the design (*e.g.*, to see if certain variables remained bounded). The engineer should know if the plant is *controllable* [9] (otherwise the control inputs will not be able to properly affect the plant) and *observable* (to see if the chosen sensors will allow the controller to observe the critical plant behavior so that it can be compensated for) or if it is *non-minimum phase*. These properties would have a fundamental impact on our ability to design effective controllers for the system. In addition, the engineer will try to make a general assessment of how the plant dynamics change over time, and what random effects are present. This analysis of the behavior of the plant gives the control engineer a fundamental understanding of the plant dynamics that will be very useful when the time comes to synthesize the controller.

1.9.2 Performance Objectives and Design Constraints

Controller design entails constructing a controller to meet the performance specifications. Often the first issue to address is whether to use open-or closed-loop control. If you can achieve your objectives with open-loop control (for example, position control using a stepper motor), why turn to feedback control? Often, you need to pay for a sensor for the feedback information and there should be justification for this cost. Moreover, feedback can destabilize this system. One should not develop a feedback control just because one is used to do it, in some simple cases an open-loop controller may provide adequate performance.

Assuming that feedback control is used the closed-loop specifications (or "performance objectives") can involve the following issues:

(*i*) **Disturbance rejection properties:** (with reference to the process fluid temperature control problem in Fig. 1.3, the control system should be able to minimize the variations in the process flow rate F). Basically, the need for minimization of the effect of

disturbance make the feedback control superior to open-loop control; for many systems it is simply impossible to meet the specifications without feedback (*e.g.,* for the temperature control problem, if you had no measurement of process fluid, how well could you relate the temperature to the user's set-point ?).

(*ii*) **Insensitivity to plant parameter variations:** (*e.g.,* for the process fluid control problem that the control system will be able to compensate for changes in the level of the process fluid in the kettle, or its thermal constant).

(*iii*) **Stability:** (*e.g.,* in the control system of Fig. 1.3, to guarantee that in absence of flow rate disturbances and change in the ambient conditions, the fluid temperature will be equal to the desired set point).

(*iv*) **Rise-time:** (*e.g.,* in the control system of Fig. 1.3, a measure of how long it takes for the actual temperature to get close to the desired temperature when there is a step change in the set-point).

(*v*) **Overshoot:** (when there is a step change in the set point in Fig. 1.3, how much the temperature will increase above the set point).

(*vi*) **Settling time:** (*e.g.*, in the control system of Fig. 1.3, how much time it takes for the temperature to reach to within a pre-assigned percent (2 or 5%) of the set point).

(*vii*) **Steady-state error:** (in absence of any disturbance in the control system of Fig. 1.3, whether the error between the set-point and actual temperature will become zero).

Apart from these technical issues, there are other issues to be considered that are often of equal or greater importance. These include:

(*i*) **Cost:** How much money and time will be needed to implement the controller?

(*ii*) **Computational complexity:** When a digital computer is used as a controller, how much processor power and memory will it take to implement the controller?

(*iii*) **Manufacturability:** Does your controller has any extraordinary requirements with regard to manufacturing the hardware that is required to implement it (*e.g.*, solar power control system)?

(*iv*) **Reliability:** Will the controller always perform properly? What is its 'meantime between failures'?

(*v*) **Maintainability:** Will it be easy to perform maintenance and routine field adjustments to the controller?

(*vi*) **Adaptability:** Can the same design be adapted to other similar applications so that the cost of later designs can be reduced ? In other words, will it be easy to modify the process fluid temperature controller to fit on different processes so that the development can be just once?

(*vii*) **Understandability:** Will the right people be able to understand the approach to control? For example, will the people that implement it or test it be able to fully understand it?

(*viii*) **Politics:** Is your boss biased against your approach? Can you sale your approach to your colleagues? Is your approach too novel (solar power control!) and does it thereby depart too much from standard company practice?

The above issues, in addition to meeting technical specifications, must also be taken into consideration and these can often force the control engineer to make decisions that can significantly affect how, for example, the ultimate process fluid controller is designed. It is important then that the engineer has these issues in mind at early stages of the design process.

1.9.3 Controller Design

Conventional control has provided numerous methods for realizing controllers for dynamic systems. Some of these are:

(*i*) *Proportional-integral-derivative (PID) control:* Over 90 % of the controllers in operations today are PID controllers (or some variation of it like a P or PI). This approach is often viewed as simple, reliable, and easy to understand. Often, like fuzzy controllers, heuristics are used to tune PID controllers (*e.g.,* the Zeigler-Nichols tuning rules).

(*ii*) *Classical control:* lead-lag compensation, Bode and Nyquist methods, root-locus design, and so on.

(*iii*) *State-space methods:* State feedback, observers, and so on.

(*iv*) *Optimal control:* Linear quadratic regulators, use of Pontryagin's minimum principle or dynamic programming, and so on.

(*v*) *Robust control:* H_2 or H *infinity* methods, quantitative feedback theory, loop shaping, and so on.

(*vi*) *Nonlinear methods:* Feedback linearization, Lyapunov redesign, sliding mode control, backstepping, and so on.

(*vii*) *Adaptive control:* Model reference adaptive control, self-tuning regulators, nonlinear adaptive control, and so on.

(*viii*) *Discrete event systems:* Petri nets, supervisory control, Infinitesimal perturbation analysis and so on.

If the engineers do not fully understand the plant and just take the mathematical model as such, it may lead to development of unrealistic control laws.

1.9.4 Performance Evaluation

The next step in the design process is the system analysis and performance evaluation. The performance evaluation is an essential step before *commissioning* the control system to check if the designed controller really meets the closed-loop specifications. This can be particularly important in safety-critical applications such as a nuclear power plant control or in aircraft control. However, in some consumer applications such as the control of washing machine or an electric shaver, it may not be as important in the sense that failures will not imply the loss of life (just the possible embarrassment of the company and cost of warranty expenses), so some rigorous evaluation matters can sometimes be ignored. Basically, there are three general ways to verify that a control system is operating properly (1) mathematical analysis based on the use of formal models, (2) simulation based analysis that most often uses formal models, and (3) experimental investigations on the real system.

(*a*) **Reliability of mathematical analysis.** In the analysis phase one may examine the stability (asymptotically stable, or bounded-input bounded-output (BIBO) stable) and controllability of the system together with other closed-loop specifications such as disturbance rejection, rise-time, overshoot, settling time, and steady-state errors. However, one should not forget the limitations of mathematical analysis. Firstly, the accuracy of the analysis is no better than that of the mathematical model used in the analysis, which is never a perfect representation of the actual plant, so the conclusions that are arrived at from the analysis are in a sense only as accurate as the model itself. And, secondly, there is a need for the development of analysis techniques for even more sophisticated nonlinear systems since existing theory is somewhat lacking for the analysis of complex nonlinear (*e.g.*, fuzzy) control systems, a large number of inputs and outputs, and stochastic effects. In spite of these limitations, the

mathematical analysis does not become a useless exercise in all the cases. Sometimes it helps to uncover fundamental problems with a control design.

(*b*) **Simulation of the designed system.** In the next phase of analysis, the controller and the actual plant is simulated on analog or digital computer. This can be done by using the laws of the physical world to develop a mathematical model and perhaps real data can be used to specify some of the parameter of the model obtained via system identification or direct parameter measurement. The simulation model can often be made quite accurate, and the effects of implementation considerations such as finite word-length restrictions in digital computer realization can be included. Currently simulations are done on digital computers, but there are occasions where an analog computer is still quite useful, particularly for real-time simulation of complex systems or in certain laboratory settings.

Simulation (digital, analog or hybrid) too has its limitations. First, as with the mathematical analysis, the model that is developed will never be identical with the actual plant. Besides, some properties simply cannot be fully verified through simulation studies. For instance, it is impossible to verify the asymptotic stability of an ordinary differential equation through simulations since a simulation can only run for a finite amount of time and only a finite number of initial conditions can be tested for these finite-length trajectories. But, simulation-based studies can provide valuable insights needed to redesign the controller before investing more time for the implementation of the controller apart from enhancing the engineer's confidence about the closed loop behavior of the designed system.

(*c*) **Experimental studies.** In the final stage of analysis, the controller is implemented and integrated with the real plant and tested under various conditions. Obviously, implementations require significant resources (*e.g.,* time, hardware), and for some plants implementation would not even be thought of before extensive mathematical and simulation-based studies have been completed. The experimental evaluation throws some light on some other issues involved in control system design such as cost of implementation, reliability, and perhaps maintainability. The limitations of experimental evaluations are, first, problems with the repeatability of experiments, and second, variations in physical components, which make the verification only approximate for other plants that are manufactured at other times. Experimental studies, also, will go a long way toward enhancing the engineer's confidence after seeing one real system in operation.

There are two basic reasons to choose one or all three of the above approaches of performance evaluation. Firstly, the engineer wants to verify that the designed controller will perform properly. Secondly, if the closed-loop system does not perform properly, then the analysis is expected to reveal a way for undertaking a redesign of the controller to meet the specifications.

CHAPTER

2

Mathematical Preliminaries

In this chapter we shall discuss briefly some of the mathematical tools used extensively for the analysis and design of control systems.

2.0 THE LAPLACE TRANSFORM

The Laplace transform method is a very useful mathematical tool [10-11] for solving linear differential equations. By use of Laplace transforms, operations like differentiation and integration can be replaced by algebraic operations such that, a linear differential equation can be transformed into an algebraic equation in a complex variable s. The solution of the differential equation may be found by inverse Laplace transform operation simply by solving the algebraic equation involving the complex variable s.

Analysis and design of a linear system are also carried out in the s-domain without actually solving the differential equations describing the dynamics of the system. Graphical techniques like Bode plot, Root Locus and Nyquist Diagram employ the Laplace transform method for evaluation of the system performance. Besides, the transient response as well as the steady-state response of a dynamic system can be obtained by the Laplace transform method.

2.1 COMPLEX VARIABLES AND COMPLEX FUNCTIONS

A complex number system is a two dimensional number system having a real part and an imaginary part. In the case of Laplace transformation, we use the notation s to represent the complex number and written as $s = \sigma + j\omega$, where σ is the real part and ω is the imaginary part. If the real part and/or imaginary part are variables, the complex number s is called a complex variable.

2.1.1 Complex Function

Any function of a complex variable will also be a complex function having real and imaginary parts. A complex function G(s) may be expressed in terms of its real and imaginary components as :

$$G(s) = A + jB$$

where A and B themselves are real quantities. The magnitude of $G(s)$ is $\| G(s) \| = \sqrt{A^2 + B^2}$, and the angle θ of $G(s)$ is $\tan^{-1}$ (B/A). The angle is considered positive when measured in the anticlockwise direction from the positive real axis. Associated with each complex function is a complex conjugate function. The complex conjugate of $G(s)$ is given by $\overline{G}(s) = A - jB$.

If a complex function $G(s)$ together with its derivatives exist in a region, it is said to be *analytic* in that region.

The function $G_1(s)$ given by :

$$G_1(s) = \frac{1}{s+2} \text{ with its derivative } \frac{d}{ds}\left(\frac{1}{s+2}\right) = -\frac{1}{(s+2)^2}$$

is found to exist everywhere, except at $s = -2$, so it is an analytic function at all points in the s plane except the point $s = -2$.

All the points in the s plane at which the complex function G(s) is found to be analytic are called *ordinary* points, whereas the points at which it is not analytic are called *singular* points. The terms *pole* and *zero* are used to describe two different types of singular points. The singular points are called *poles* of the complex function G(s) if the function or its derivatives approach infinity at these points. The function $G_1(s)$ in the previous example has a pole at $s = -2$.

A complex function G(s) is said to have a pole of order n at $s = -a$ if the product $(s + a)^n$ G(s), n = 1, 2, 3, ... has a finite, nonzero value at the pole $s = -a$.

The singular points at which the function G(s) equals zero are called *zeros*. The complex function $G_2(s)$ given by

$$G_2(s) = \frac{K(s+4)(s+6)}{s(s+1)(s+10)^2}$$

has zeros at $s = -4$, $s = -6$, simple poles at $s = 0$, $s = -1$, and a pole of order 2 at $s = -10$. It is to be noted that the function $G_2(s)$ becomes zero as $s \to \infty$. Since for large values of s it can be written as :

$$G_2(s) = \frac{K}{s^2}$$

So, $G_2(s)$ has *zeros* of order two at $s = \infty$. The complex function G(s) has the same number of *poles* as the number of *zeros*, if points at infinity are counted. So, the complex function $G_2(s)$ has four zeros at $s = -4$, $s = -6$, $s = \infty$, $s = \infty$ and four poles at $s = 0$, $s = -1$, $s = -10$ and $s = -10$.

2.2 LAPLACE TRANSFORMATION

We shall now present a definition of the Laplace transformation followed by a discussion on the existence of the Laplace transform. The derivation of Laplace transforms will be illustrated by considering some typical examples.

Let $f(t)$ be a function of time t such that $f(t) = 0$ for $t < 0$ and s is a complex variable. We use £ as a symbolic operator to stand for Laplace integral on the quantity that is prefixed by it.

$$£ \Rightarrow \text{stands for the Laplace integral } \int_o^\infty e^{-st}\, dt$$

Definition 2.1 The Laplace transform of $f(t)$ is defined as

$$£\,[f(t)] = F(s) = \int_o^\infty e^{-st}\, dt\, [f(t)] = \int_o^\infty f(t) e^{-st}\, dt \qquad (2.1)$$

where F(s) denotes the Laplace transform of $f(t)$. The function $f(t)$ in time domain can be found from the Laplace transform F(s) by the reverse process known as the *inverse Laplace transformation.* The inverse Laplace transformation is denoted by the symbol $£^{-1}$ for convenience. So, we can write

$$£^{-1}\,[F(s)] = f(t)$$

2.2.1 Laplace Transform and Its Existence

The existence of Laplace transformation of a function $f(t)$ depends on the convergence of the Laplace integral. The integral (2.1) will converge if the function $f(t)$ is piecewise continuous in every finite interval of its argument $t > 0$ and if the function is of exponential order as t approaches infinity. A function $f(t)$ is said to be of exponential order if there exists a real positive constant σ such that the absolute value of the function, multiplied by the exponential term $e^{-\sigma t}$ (that is, $e^{-\sigma t} \mid f(t) \mid$) approaches zero as t approaches infinity.

The particular value of $\sigma = \sigma_c$ is called the abscissa of convergence if the limiting value of $e^{-\sigma t} \mid f(t) \mid$ converges to zero for $\sigma < \sigma_c$ and diverges to infinity for $\sigma > \sigma_c$. That is, if the real part of s is greater than the abscissa of convergence σ_c, then the integral $\int_o^{\infty} f(t)e^{-st}\,dt$ will converge to a finite value and the Laplace transform of $f(t)$ will exist.

The abscissa of convergence σ_c , may be associated with the real part of the pole of F(s) that lie farthest to the right in the s plane. For instance, in the following function F(s),

$$F(s) = \frac{K(s+4)}{(s+2)(s+5)}$$

the abscissa of convergence σ_c is equal to –2. For functions like e^{-ct} , te^{-ct}, e^{-ct} sin ωt, the abscissa of convergence is equal to $-c$, whereas for functions like t, sin ωt, and t sin ωt the abscissa of convergence is equal to zero. It is not possible to find suitable values of the abscissa of convergence for functions that increase faster than the exponential $e^{-\sigma t}$ term. Therefore, functions like e^{t^2} and te^{t^2} do not possess Laplace transforms for any arbitrarily large value of t. They do possess a Laplace transform if t is restricted to a finite interval $0 \le t \le T$ as defined below

$$\begin{aligned} f(t) &= e^{t^2} && \text{for } 0 \le t \le T < \infty \\ &= 0 && \text{for } t < 0,\ t > T \end{aligned}$$

Such a signal can be physically generated and the signals that can be physically generated always have corresponding Laplace transforms.

2.3 LAPLACE TRANSFORM OF COMMON FUNCTIONS

We shall now illustrate the derivation of Laplace transforms for a few common functions, starting from the defining Equation (2.1).

(a) **Exponential function.** Let us consider the exponential function

$$\begin{aligned} f(t) &= ae^{-\alpha t} && \text{for } t \ge 0 \\ &= 0 && \text{for } t < 0; \text{ with constant } a \text{ and } \alpha. \end{aligned}$$

The Laplace transform of this exponential function can be obtained from the defining relation (2.1) as follows :

$$F(s) = \pounds\ [ae^{-\alpha t}] = \int_0^{\infty} ae^{-\alpha t}\, e^{-st}\, dt = a\int_0^{\infty} e^{-(\alpha+s)t}\, dt = \frac{a}{s+\alpha}$$

The exponential function in time is found to produce a pole for the complex function F(s).

Although it is required in deriving the Laplace transform of $f(t) = ae^{-\alpha t}$, that the real part of s should be greater than the abscissa of convergence, $-\alpha$, the Laplace transform computed is valid in the range where $\sigma < -\alpha$ in the s plane where it is analytic. This can be proved by using what is known as *analytic extension theorem* for complex variables.

Analytic extension theorem. *Two analytic functions of complex variables are equal everywhere in the region if they are equal for a finite length along any arc in the region in which both are analytic.*

Normally the real axis or a part of it is taken as the arc of equality. The Laplace transform F(s) is first found by performing the integration in which s is allowed to have a positive real value greater than the abscissa of convergence. It will be analytic on portions of real axis excluding the poles of F(s). By using the extension theorem, F(s) will be considered valid in the entire s plane where it is analytic, even though it was required that the real values of s should be greater than the abscissa of convergence for the existence of Laplace integral.

(b) **Step function.** Let us Consider the step function

$$f(t) = \mathrm{R}\, u_s(t) \qquad \text{for } t \geq 0$$
$$= 0 \qquad \text{for } t < 0 \tag{2.2}$$

where R is a constant and $u_s(t)$ is an unit step function. The step function may be considered a special case of the exponential function $\mathrm{R}e^{-\alpha t}$, with $\alpha = 0$. The *unit-step* function has a height of unity. The unit-step function that occurs at $t = t_o$ is written as $u_s(t - t_o)$. The step function of height R can then be written as $f(t) = \mathrm{R}\, u_s(t) = \mathrm{R} \,.\, 1(t)$ (see Fig. 2.1).

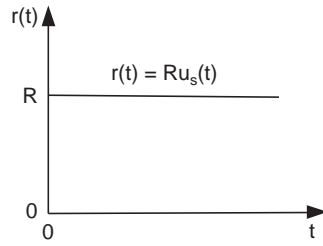



Fig. 2.1 Step function

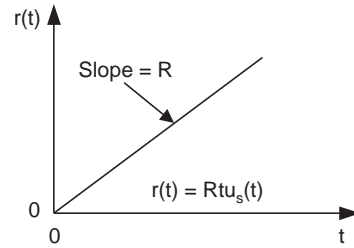



Fig. 2.2 Ramp function

The Laplace transform of step function with strength or height R is given by

$$£\,[\mathrm{R}] = \int_0^{\infty} \mathrm{R}u_s(t)\, e^{-st}\, dt = \frac{\mathrm{R}}{s} \tag{2.3}$$

In performing this integration, we assumed that the real part of s was greater than zero (the abscissa of convergence) such that the lim e^{-st} was zero. However, the Laplace transform given by Equation (2.3) is considered valid in the entire s plane except at the pole $s = 0$. Physically, a step function occurring at $t = 0$ corresponds to a constant signal suddenly applied to a system at time t equal to zero.

(c) **Ramp function.** The ramp function with slope R is given by

$$f(t) = \mathrm{R}t\, u_s(t) \qquad \text{for } t \geq 0$$
$$= 0 \qquad \text{for } t < 0 \tag{2.4}$$

where R is a constant. The plot of ramp function is shown in Fig. 2.2. The Laplace transform of this ramp function is found as

$$£\,[\mathrm{R}t] = \int_0^{\infty} \mathrm{R}tu_s(t)\, e^{-st}\, dt = \mathrm{R}t\, \frac{e^{-st}}{-s}\Bigg|_0^{\infty} - \int_0^{\infty} \frac{\mathrm{R}u_s(t)\, e^{-st}}{-s}\, dt$$

$$= \frac{\mathrm{R}}{s}\int_0^{\infty} e^{-st}\, dt = \frac{\mathrm{R}}{s^2}. \tag{2.5}$$

(*d*) **Sinusoidal function.** Let us consider the sinusoidal function

$$\begin{aligned} f(t) &= \text{A} \sin \omega t \qquad \text{for } t \geq 0 \\ &= 0 \qquad\qquad \text{for } t < 0 \end{aligned} \tag{2.6}$$

where A and ω are constants. We note that sin ω*t* can be written as:

$$\sin \omega t = \frac{1}{2j} (e^{j\omega t} - e^{-j\omega t})$$

Hence, the Laplace transform of A sin ω*t* is found as

$$£ [\text{A} \sin \omega t] = \frac{\text{A}}{2j} \int_0^\infty (e^{j\omega t} - e^{-j\omega t})\, e^{-st}\, dt$$

$$= \frac{\text{A}}{2j} \frac{1}{s - j\omega} - \frac{\text{A}}{2j} \frac{1}{s + j\omega} = \frac{\text{A}\omega}{s^2 + \omega^2} \tag{2.7}$$

Proceeding in a similar way, the Laplace transform of A cos ω*t* can be found to be :

$$£ [\text{A} \cos \omega t] = \frac{\text{A}s}{s^2 + \omega^2}$$

(*e*) **Pulse function.** Consider the pulse function shown in Fig. 2.3,

$$\begin{aligned} f(t) &= \frac{\text{A}}{\text{T}} \qquad \text{for } 0 < t < \text{T} \\ &= 0 \qquad \text{for } t < 0,\ t > \text{T} \end{aligned}$$

where A and T are constants.

Fig. 2.3 A Pulse of duration T

The pulse function in Fig. 2.3 may be considered to consist of two step functions: one of height $\frac{\text{A}}{\text{T}}$ applied at $t = 0$ superimposed with another negative step function of height $\frac{\text{A}}{\text{T}}$ applied $t = \text{T}$; such that it may be written as,

$$f(t) = \frac{\text{A}}{\text{T}} u_s(t) - \frac{\text{A}}{\text{T}} u_s(t - \text{T}) \tag{2.8}$$

The Laplace transform of the pulse *f*(*t*) can, therefore, be found as

$$£ [f(t)] = £ \left[\frac{\text{A}}{\text{T}} u_s(t)\right] - £ \left[\frac{\text{A}}{\text{T}} u_s(t - \text{T})\right] = \frac{\text{A}}{\text{T}s} - \frac{\text{A}}{\text{T}s} e^{-s\text{T}} = \frac{\text{A}}{\text{T}s} (1 - e^{-s\text{T}}) \tag{2.9}$$

(*f*) **Impulse function.** An impulse function of strength $\frac{\text{A}}{\text{T}}$ and duration T is defined as

$$\begin{aligned} f(t) &= \lim_{\text{T} \to o} \frac{\text{A}}{\text{T}} \qquad \text{for } 0 < t < \text{T} \\ &= 0 \qquad\qquad \text{for } t < 0,\ t > \text{T} \end{aligned} \tag{2.10}$$

It is a limiting case of the pulse function as the duration T approaches zero.

The strength of an impulse is represented by its area. The strength of the impulse represented by the relation (2.10) is A, since the height of the impulse function is $\frac{\text{A}}{\text{T}}$ and the duration is T.

Now, as the duration T is allowed to approach zero, the height $\frac{A}{T}$ approaches infinity, but the area under the impulse and hence the strength remains unchanged. The Laplace transform of the impulse function may be found by applying the limiting condition to the Laplace transform of pulse represented by Equation (2.9).

$$£ f(t) = \lim_{T \to 0} \left[\frac{A}{Ts} (1 - e^{-sT}) \right]$$

Using the rule of L Hospital, we get, $£ f(t) = \lim_{T \to o} \dfrac{\frac{d}{dT}[A(1-e^{-sT})]}{\frac{d(Ts)}{dT}} = \frac{As}{s} = A$ (2.11)

So, the Laplace transform of the impulse function is found to be equal to its strength.

The impulse function with *unity strength* is called by a special name by the Physicists. They call it the *Dirac delta function*. The *Dirac delta function* occurring at $t = t_o$ is denoted by $\delta(t - t_o)$. It has following properties :

$$f(t - t_o) = \delta(t - t_o) = 0 \quad \text{for } t \neq t_o$$
$$f(t - t_o) = \delta(t - t_o) = \infty \quad \text{for } t = t_o$$

$$\int_{-\infty}^{\infty} \delta(t - t_o)\, dt = 1(t) = u_s(t)$$

Needless to say, an impulse function with infinite magnitude and zero duration can not be realized in practice and is only a mathematical concept. However, if a pulse of large magnitude whose duration is very short compared to the smallest time constant of the system, is impressed to a system, then the pulse input may be approximated by an impulse function. For example, in the case of a ceiling fan, if power switch is turned on and off once in quick succession, then the input power to the fan may be approximated by an impulse and the response of the fan will be due to an impulse.

Differentiation of discontinuous functions can be performed by employing the concept of the impulse function. For example, the unit-impulse function $\delta(t - t_o)$ can be considered to be the derivative of the unit-step function $u_s(t - t_o)$ at the point of discontinuity $t = t_o$ as shown below:

$$\delta(t - t_o) = \frac{d}{dt}\, u_s(t - t_o)$$

Conversely, it may be thought that the unit-step function $u_s(t - t_o)$ is the result of integration of the unit-impulse function $\delta(t - t_o)$.

Using the concept of the impulse function, differentiation of any function with discontinuities can be performed. The differentiation will produce impulses whose magnitudes are numerically equal to the magnitude of the function at the point of discontinuity.

2.3.1 Laplace Table

The Laplace transforms for functions $f(t)$ commonly encountered in connection with the study of a subject may be found once for all by multiplying $f(t)$ by e^{-st} and then integrating the product from $t = 0$ to $t = \infty$ and the resulting transformations be kept in a Table for future reference. The Laplace transforms of time functions that are encountered more frequently in

the analysis of *linear systems,* including *control systems*, are recorded in Table 2.1 for easy reference.

In what follows, we shall present Laplace transforms of time functions along with some theorems, without proof, on the Laplace transformation that are relevant in the study of linear control systems. The theorems can be proved easily with the help of defining Equation (2.1).

2.4 PROPERTIES OF LAPLACE TRANSFORM

Some properties of the Laplace Transform are listed below in the form of theorems and can be easily proved by using the definition of Laplace Transform in Equation (2.1) and properties of complex variables.

Theorem 2.1 The Laplace transform of a.f(t), where a is constant is given by

$$£\ [af(t)] = a£\ [f(t)]$$

This is obvious from the definition of the Laplace transform. Similarly, if functions $f_1(t)$ and $f_2(t)$ have Laplace transforms, then the Laplace transform of the function $f_1(t) + f_2(t)$ is given by

$$£\ [f_1(t) + f_2(t)] = £\ [f_1(t)] + £\ [f_2(t)]$$

Theorem 2.2 Multiplication of f(t) by $e^{-\alpha t}$

If $f(t)$ has Laplace transform F(s), then the Laplace transform of $e^{-at} f(t)$ is obtained as

$$£\ [e^{-\alpha t} f(t)] = \int_o^{\infty} e^{-\alpha t} f(t) e^{-st}\, dt = \mathrm{F}(s + \alpha) \tag{2.12}$$

We note that the multiplication of $f(t)$ by $e^{-\alpha t}$ has the effect of replacing s by $(s + \alpha)$ in the Laplace transform F(s). Conversely, changing s to $(s + \alpha)$ has the same effect as of multiplying $f(t)$ by $e^{\alpha t}$ in time domain. Here α may be real or complex number.

Theorem 2.3 Translation in time

When a function $f(t)\, u_s(t)$ is translated in time by $\alpha \geq 0$ to become $f(t - \alpha)\, u_s(t - \alpha)$, the Laplace transform of the translated function is given by $e^{-\alpha s}$ F(s) where F(s) is the Laplace transform of $f(t)$.

Theorem 2.4 Change of time scale

If the independent variable t is scaled to t/α, where α is a positive constant, then the function $f(t)$ is modified as $f(t/\alpha)$. The time-scaled function $f(t/\alpha)$ will be faster compared to $f(t)$ if $\alpha > 1$ and slower if $\alpha < 1$. We need time scaling in solving a differential equation which is very slow or very fast compared to the unit of time used to observe the solution. For example, we can not sit for 365 days before a computer to record the motion of earth around the sun as the solution of a differential equation or record the motion of an electron on a plotter without time-scaling.

Denoting the Laplace transform of $f(t)$ by F(s), the Laplace transform of $f(t/\alpha)$ will be given by:

$$£\left[f\left(\frac{t}{\alpha}\right)\right] = \int_o^{\infty} f\left(\frac{t}{\alpha}\right) dt = \alpha \mathrm{F}\ (\alpha s)$$

In order to illustrate the point in Theorem 2.4, we note that $£\ [e^t] = \mathrm{F}(s) = \frac{1}{s+1}$. Therefore, with $t/20 = 0.05t$, $£\ [e^{0.05t}] = 20\ \mathrm{F}(20s) = \frac{20}{20s+1}$. The same result will be obtained by using the defining relation (2.1) for $f(t) = e^{0.05t}$.

Theorem 2.5 Real differentiation

When the Laplace transform F(s) of a function $f(t)$ is available, the Laplace transform of the derivative of the function $f(t)$ can be obtained in terms of F(s) as

$$£\left[\frac{d}{dt}f(t)\right] = sF(s) - f(0) \tag{2.13}$$

where $f(0)$ is the value of $f(t)$ evaluated at $t = 0$. When $f(t)$ has a discontinuity at $t = 0$, the left hand limit $f(0_-)$ is different from right hand limit $f(0_+)$, in which case Equation (2.13) must be modified to

$$£_+\left[\frac{d}{dt}f(t)\right] = sF(s) - f(0_+) \quad \text{and} \quad £_-\left[\frac{d}{dt}f(t)\right] = sF(s) - f(0_-)$$

In general, for the nth derivative of $f(t)$, we can have

$$£\left[\frac{d^n}{dt^n}f(t)\right] = s^n F(s) - s^{n-1} f(0) - s^{n-2} f'(0) - \ldots - sf^{(n-2)}(0) - f^{(n-1)}(0)$$

where $f(0)$, $f'(0)$, $f^{(n-1)}(0)$ represent the values of $f(t)$, $df(t)/dt$, ….. $d^{n-1} f(t)/dt^{n-1}$, respectively, evaluated at $t = \mathbf{0}$.

Theorem 2.6 Initial value

The initial value theorem may be stated as follows. If the Laplace transform of both $f(t)$ and $df(t)/dt$ exist together with the $\lim_{s\to\infty} sF(s)$, then

$$f(0_+) = \lim_{s\to\infty} sF(s)$$

By using the initial value theorem, we can find the value of the function $f(t)$ at time $t = 0_+$ directly from the Laplace transform F(s), without taking the *inverse* Laplace transformation.

The initial value theorem and the final value theorem (stated below) are very useful for assessing the steady state performance of a stable control system directly from its transfer function in frequency domain without actually transforming functions in s back to time domain.

Theorem 2.7 Final value

If the Laplace transform of both $f(t)$ and $df(t)/dt$ exist together with the $\lim_{t\to\infty} f(t)$, then the final value of the function $f(t)$ is obtained as

$$\lim_{t\to\infty} f(t) = \lim_{s\to 0} sF(s), \text{ where F}(s) \text{ is the Laplace transform of } f(t).$$

The final value theorem provides a convenient means for computing the steady state behavior of a linear dynamic system in time domain without explicitly solving for the function in time domain by inverse Laplace transform. However, the final value theorem is applicable only when the $\lim_{t\to\infty} f(t)$ exists, for which the poles of sF(s) must lie on the left hand side of the $j\omega$ axis in the s-plane (that is poles must have negative real parts). If this is not the case, the conclusion arrived at from the final value theorem will be wrong.

Table 2.1. Laplace Transform Pairs

	$f(t)$	$F(s)$
1.	Unit impulse $\delta(t)$	1
2.	Unit step 1 $u_s(t)$	$\frac{1}{s}$
3.	t	$\frac{1}{s^2}$
4.	$\frac{t^{n-1}}{(n-1)!}$ $(n = 1, 2, 3, ...)$	$\frac{1}{s^n}$
5.	t^n $(n = 1, 2, 3, ...)$	$\frac{n!}{s^{n+1}}$
6.	e^{-at}	$\frac{1}{s+a}$
7.	te^{-at}	$\frac{1}{(s+a)^2}$
8.	$\frac{1}{(n-1)!} t^{n-1} e^{-at}$ $(n = 1, 2, 3, ...)$	$\frac{1}{(s+a)^n}$
9.	$t^n e^{-at}$ $(n = 1, 2, 3, ...)$	$\frac{n!}{(s+a)^{n+1}}$
10.	$\sin \omega t$	$\frac{\omega}{s^2+\omega^2}$
11.	$\cos \omega t$	$\frac{s}{s^2+\omega^2}$
12.	$\sinh \omega t$	$\frac{\omega}{s^2-\omega^2}$
13.	$\cosh \omega t$	$\frac{s}{s^2-\omega^2}$
14.	$\frac{1}{a}(1-e^{-at})$	$\frac{1}{s(s+a)}$
15.	$\frac{1}{b-a}(e^{-at}-e^{-bt}), b \neq a$	$\frac{1}{(s+a)(s+b)}$
16.	$\frac{1}{b-a}(be^{-bt}-ae^{-at}), b \neq a$	$\frac{s}{(s+a)(s+b)}$
17.	$\frac{1}{ab}\left[1+\frac{1}{a-b}(be^{-at}-ae^{-bt})\right], b \neq a$	$\frac{1}{s(s+a)(s+b)}$
18.	$\frac{1}{a^2}(1-e^{-at}-ate^{-at})$	$\frac{1}{s(s+a)^2}$
19.	$\frac{1}{a^2}(at-1+e^{-at})$	$\frac{1}{s^2(s+a)}$

20.	$e^{-at}\sin\omega t$	$\dfrac{\omega}{(s+a)^2+\omega^2}$
21.	$e^{-at}\cos\omega t$	$\dfrac{s+a}{(s+a)^2+\omega^2}$
22.	$\dfrac{\omega_n}{\sqrt{1-\delta^2}}\, e^{-\delta\omega_n t}\sin\omega_n(\sqrt{1-\delta^2})t,\ \delta<1$	$\dfrac{\omega_n^2}{s^2+2\delta\omega_n s+\omega_n^2}$
23.	$-\dfrac{1}{\sqrt{1-\delta^2}}\, e^{-\delta\omega_n t}\sin\{\omega_n(\sqrt{1-\delta^2})t-\varphi\},\ \delta<1$; $\varphi=\tan^{-1}\left(\dfrac{\sqrt{1-\delta^2}}{\delta}\right)$	$\dfrac{s}{s^2+2\delta\omega_n s+\omega_n^2}$
24.	$1-\dfrac{1}{\sqrt{1-\delta^2}}\, e^{-\delta\omega_n t}\sin\{\omega_n(\sqrt{1-\delta^2})t+\varphi\},\ \delta<1$; $\varphi=\tan^{-1}\left(\dfrac{\sqrt{1-\delta^2}}{\delta}\right)$	$\dfrac{\omega_n^2}{s(s^2+2\delta\omega_n s+\omega_n^2)}$
25.	$1-\cos\omega t$	$\dfrac{\omega^2}{s(s^2+\omega^2)}$
26.	$\omega t-\sin\omega t$	$\dfrac{\omega^3}{s^2(s^2+\omega^2)}$
27.	$\sin\omega t-\omega t\cos\omega t$	$\dfrac{2\omega^3}{(s^2+\omega^2)^2}$
28.	$\dfrac{1}{2\omega}t\sin\omega t$	$\dfrac{s}{(s^2+\omega^2)^2}$
29.	$t\cos\omega t$	$\dfrac{s^2-\omega^2}{(s^2+\omega^2)^2}$
30.	$\dfrac{1}{\omega_2^2-\omega_1^2}(\cos\omega_1 t-\cos\omega_2 t)\ ;\ (\omega_1^2\neq\omega_2^2)$	$\dfrac{s}{(s^2+\omega_1^2)(s^2+\omega_2^2)}$
31.	$\dfrac{1}{2\omega}(\sin\omega t+\omega t\cos\omega t)$	$\dfrac{s^2}{(s^2+\omega^2)^2}$

Theorem 2.8 Real integration

If the Laplace transform F(*s*) of $f(t)$ is available, then the Laplace transform of the integral of the function, $\int f(t)\,dt$, can be expressed in terms of F(s) as follows:

$$£\,[\int f(t)\,dt]=\frac{F(s)}{s}+\frac{f^{-1}(0)}{s} \tag{2.14}$$

where $F(s)=£_+[f(t)]$ and $f^{-1}(0)=\int f(t)\,dt$, evaluated at $t=0$.

Since, for an impulse function the left and right hand side limits are different at $t = 0$, *i.e.*, $f^{-1}(0_+) \neq f^{-1}(0_-)$, if $f(t)$ happens to be an impulse function, the Equation (2.14) should be modified as follows:

$$£_+ [\int f(t)\, dt] = \frac{F(s)}{s} + \frac{f^{-1}(0_+)}{s}$$

and

$$£_- [\int f(t)\, dt] = \frac{F(s)}{s} + \frac{f^{-1}(0_-)}{s}$$

We note that integration of a function in time domain is reflected as a division of F(s) by s in the s-domain. In the case of definite integral of $f(t)$, Equation (2.14) will be modified as follows :

If the Laplace transform of $f(t)$ exists and is denoted by F(s), the Laplace transform of the definite integral

$$\int_0^t f(t)\, dt \text{ is given by } £\left[\int_0^t f(t)\, dt\right] = \frac{F(s)}{s} \qquad (2.15)$$

Theorem 2.9 Complex differentiation

If F(s) denotes the Laplace transform of $f(t)$, then the Laplace transform of $tf(t)$ is given by $£ [tf(t)\, dt] = -\frac{d}{ds} F(s)$, at all values of s excluding the poles of F(s). This theorem is known as the complex differentiation theorem.

The complex differentiation theorem in general form is given by:

$$£ [t^n f(t)\, dt] = (-1)^n \frac{d^n}{ds^n} F(s) \; (n = 1, 2, 3, ...)$$

Theorem 2.10 Convolution integral

If $f_1(t)$ and $f_2(t)$ are piecewise continuous time functions having Laplace transforms $F_1(s)$ and $F_2(s)$ respectively, then the Laplace transform of the integral $\int_0^t f_1(t-\tau) f_2(t)\, d\tau$ is given by

$$£\left[\int_0^t f_1(t-\tau)\, f_2(\tau)\, d\tau\right] = F_1(s)\, F_2(s) \qquad (2.16)$$

where $F_1(s) = \int_0^\infty f_1(t)\, e^{-st}\, dt = £ f_1(t)$ and $F_2(s) = \int_0^\infty f_2(t)\, e^{-st}\, dt = £ f_2(t)$

The integral on left hand side of Equation (2.16) is often written as $f_1(t) * f_2(t)$ and is called the convolution of $f_1(t)$ and $f_2(t)$.

If two functions in the s-domain appear as product of two Laplace transform functions, $F_1(s)F_2(s)$, then the inverse Laplace transform of $F_1(s)\, F_2(s)$, is given by the convolution integral $f_1(t) * f_2(t)$ in time-domain.

2.5 INVERSE LAPLACE TRANSFORMATION

The mathematical process for obtaining the time domain function from the function in the domain of complex variable s is known as *inverse Laplace transformation*, denoted by the symbol $£^{-1}$ so that $£^{-1} [F(s)] = f(t)$.

The Laplace transform is used to convert a differential and integral equation to an algebraic equation as an intermediate step for ease of manipulation. But we have to revert back to the time domain for proper physical interpretation of the solution. The function $f(t)$ is computed from F(s) by the following integral:

$$f(t) = \frac{1}{2\pi} \int_{c-j\infty}^{c+j\infty} F(s)\, e^{st}\, ds \;\; (t > 0)$$

where c is a real constant and is known as the abscissa of convergence. The constant is chosen larger than the real parts of all singular points of F(s) such that the path of integration is parallel to the $j\omega$ axis and is displaced to its left through c. The above path of integration is to the right of all singular points of F(s). The computation of the inversion integral is normally complicated. However, there are alternative and simpler methods for finding $f(t)$ from F(s). A convenient method for obtaining inverse Laplace transforms is to use a table of Laplace transforms of standard functions. The function F(s) is expanded in partial fraction so that the individual terms can be easily identified with the standard terms in a Table for which the inverse Laplace transforms are known. The one to one correspondence of a continuous time function with its inverse Laplace transform is exploited in preparing the Laplace transform Table.

2.5.1 Partial-Fraction Expansion Method

The Laplace transforms of time functions $f(t)$ encountered in the study of control systems are normally expressed as the ratio of two polynomials in s:

$$F(s) = \frac{P(s)}{Q(s)}$$

where the degree of P(s) is less than that of Q(s). It is convenient to express F(s) in terms of its partial fraction components as:

$$F(s) = F_1(s) + F_2(s) + \dots + F_n(s)$$

and identify the inverse Laplace transforms of $F_1(s)$, $F_2(s)$ ……, $F_n(s)$ with unique time functions $f_1(t)$, $f_2(t)$, ….., $f_n(t)$ from a table, so that one can write

$$£^{-1}[F(s)] = £^{-1}[F_1(s)] + £^{-1}[F_2(s)] + \dots + £^{-1}[F_n(s)] = f_1(t) + f_2(t) + \dots + f_n(t)$$

For a continuous time function f(t), there is a one-to-one mapping between f(t) and its Laplace transform F(s).

However, before writing the partial fraction expansion of F(s), the roots of the equation Q(s) = 0 must be known beforehand. In the expansion of F(s) = P(s)/Q(s) into partial-fraction form, if the highest power of s in P(s) be greater than or equal to the highest power of s in Q(s), then the numerator P(s) may be divided by the denominator Q(s) in order to produce a polynomial in s plus a remainder (vide Example 2.2).

2.5.2 Partial-Fraction Expansion when F(s) has only Distinct Poles

When F(s) has distinct poles it can be written as

$$F(s) = \frac{P(s)}{Q(s)} = \frac{K(s+z_1)(s+z_2)\dots(s+z_m)}{(s+p_1)(s+p_2)\dots(s+p_n)} \;;\; (n > m)$$

where $p_1, p_2, \dots p_n$ and $z_1, z_2 \dots.., z_m$ are distinct poles and zeros of F(s). The poles and zeros may be either real or complex and when the poles or zeros are complex they occur in conjugate pairs. Since F(s) has only distinct poles, it can be expanded into a sum of simple partial fractions as :

$$F(s) = \frac{P(s)}{Q(s)} = \frac{a_1}{s + p_1} + \frac{a_2}{s + p_2} + \ldots + \frac{a_n}{s + p_n} \tag{2.17}$$

where the constants $a_j (j = 1, 2, \ldots\ldots n)$ are called the residues at the pole at $s = -p_j$. The residue a_j is found by multiplying both sides of Equation (2.17) by $(s + p_j)$ and putting $s = -p_j$, as shown below:

$$\left[(s + p_j)\frac{P(s)}{Q(s)}\right]_{s=-p_j}$$

$$= \left[\frac{a_1}{s + p_1}(s + p_j) + \frac{a_2}{s + p_2}(s + p_j) + \ldots + a_j + \ldots \frac{a_n}{s + p_n}(s + p_j)\right]_{s=-p_j} = a_j$$

All the expanded terms on the right hand side are found to be zero except a_j. So the residue a_j is computed from the following relation.

$$a_j = (s + a_j)\left.\frac{P(s)}{Q(s)}\right|_{s=-a_j} \tag{2.18}$$

In case of complex conjugate pole pairs (p_1, p_2) only one of the associated coefficients, a_1 or a_2, is to be found and the other coefficient can be written from the first.

Since we know that $£^{-1}\left[\frac{a_j}{s + p_j}\right] = a_j e^{-p_j t}$

we can write $f(t)$ from (2.17) as $f(t) = £^{-1}\, F(s) = a_1 e^{-p_1 t} + a_2 e^{-p_2 t} + \ldots + a_n e^{-p_n t}, \quad (t \geq 0)$

Example 2.1 Let us find the time function from the following Laplace transform

$$F(s) = \frac{s + 4}{(s + 1)(s + 2)}$$

The partial-fraction expansion of F(*s*) can be written as:

$$F(s) = \frac{s + 4}{(s + 1)(s + 2)} = \frac{a_1}{s + 1} + \frac{a_2}{s + 2}$$

where the residues a_1 and a_2 are found by using Equation (2.18),

$$a_1 = \left[(s + 1)\frac{s + 4}{(s + 1)(s + 2)}\right]_{s=-1} = \left[\frac{s + 4}{s + 2}\right]_{s=-1} = 3$$

and $$a_2 = \left[(s + 2)\frac{s + 4}{(s + 1)(s + 2)}\right]_{s=-2} = \left[\frac{s + 4}{s + 1}\right]_{s=-2} = -2$$

Therefore, we have $f(t) = £^{-1}\,[F(s)] = £^{-1}\left[\frac{3}{s + 1}\right] + £^{-1}\,\frac{-2}{s + 2} = 3e^{-t} - 2e^{-2t}, \ (t \geq 0)$

Example 2.2 Let us find the inverse Laplace transform of G(*s*) given by

$$G(s) = \frac{2s^2 + 15s + 24}{s^2 + 6s + 8}$$

In this case, since the degree of the numerator polynomial is equal to that of the denominator polynomial, numerator may be divided by the denominator yielding:

$$G(s) = 2 + \frac{3s+8}{(s+2)(s+4)}$$

Using partial fraction expansion, we get $G(s) = 2 + \frac{1}{s+2} + \frac{2}{s+4}$

Hence, taking the inverse operation we get, $g(t) = 2\delta(t) + e^{-2t} + e^{-4t}$, $(t \geq 0_)$
where $\delta(t)$ is the unit impulse function.

Example 2.3 Let us find the inverse Laplace transform of F(s) given by

$$F(s) = \frac{3s+26}{s^2+4s+20}$$

The denominator polynomial can be factored as $s^2 + 4s + 20 = (s + 2 + j4)(s + 2 - j4) = (s + 2)^2 + 4^2$. Now let us look at the Laplace transform of the damped sine and cosine functions reproduced below from Table 2.1.

$$£\,[e^{-\alpha t} \sin \omega t] = \frac{\omega}{(s+\alpha)^2 + \omega^2} \quad \text{and} \quad £\,[e^{-\alpha t} \cos \omega t] = \frac{s+\alpha}{(s+\alpha)^2+\omega^2}$$

The function F(s) can be easily identified with the right hand sides of the above expressions as shown below.

$$F(s) = \frac{3s+26}{s^2+4s+20} = \frac{20+3(s+2)}{s^2+4s+20} = 5\frac{4}{(s+2)^2+4^2} + 3\frac{s+2}{(s+2)^2+4^2}$$

So, the function in time domain is found as $f(t) = £^{-1}\,[F(s)]$

$$= 5£^{-1}\left[\frac{4}{(s+2)^2+4^2}\right] + 3£^{-1}\left[\frac{s+2}{(s+2)^2+4^2}\right]$$

$$= 5e^{-2t} \sin 4t + 3e^{-2t} \cos 4t \quad (t \geq 0)$$

2.5.3 Partial-Fraction Expansion of F(s) with Repeated Poles

Suppose, for the polynomial F(s) there is pole located at $-r$ repeated q times so that F(s) may be expressed as :

$$F(s) = \frac{P(s)}{Q(s)} = \frac{P(s)}{(s+r)^q}$$

It can be expanded in partial fraction as

$$F(s) = \frac{a_1}{(s+r)^q} + \frac{a_2}{(s+r)^{q-1}} + \frac{a_3}{(s+r)^{q-2}} + \ldots + \frac{a_q}{(s+r)} \tag{2.19}$$

Multiplying both sides by $(s + r)^q$ we get

$$(s+r)^q F(s) = a_1 + a_2(s+r) + a_3(s+r)^2 + \ldots a_q(s+r)^{q-1} \tag{2.20}$$

Differentiating $(q - 1)$ times both sides of (2.20) with respect to s and computing the residue at $s = -r$, it can be shown that the coefficient a_q is given by

$$a_q = \frac{1}{(q-1)!} \frac{d^{q-1}}{ds^{q-1}} [(s+r)^q F(s)]_{s=-r} \tag{2.21}$$

We can find the residue for the first term in (2.19) using relation (2.20), as $a_1 = [(s + r)^q F(s)]_{s=-r}$ and the second term as $a_2 = \frac{d}{ds}[(s + r)^q F(s)]_{s=-r}$ and so on.

Example 2.4 Consider the following polynomial

$$F(s) = \frac{4s^2 + 19s + 24}{(s+2)^3}$$

The partial-fraction expansion of this F(*s*) involves three terms,

$$F(s) = \frac{P(s)}{Q(s)} = \frac{a_1}{(s+2)^3} + \frac{a_2}{(s+2)^2} + \frac{a_3}{s+2}$$

where a_1, a_2, and a_3 are found from (2.21) as

$$a_1 = [(s + 2)^3 F(s)]_{s=-2} = 4(-2)^2 + 19(-2) + 24 = 2;$$

$$a_2 = \frac{d}{ds}[(s + 2)^3 F(s)]_{s=-2} = (8s + 19)\,|_{s=-2} = 3$$

and

$$a_3 = \frac{1}{2!}\frac{d^2}{ds^2}[(s + 2)^3 F(s)]_{s=-2} = (1/2)8 = 4$$

Hence $F(s) = \frac{2}{(s+2)^3} + \frac{3}{(s+2)^2} + \frac{4}{(s+2)}$; We note from row 9 of the Laplace transform pairs Table 2.1 that $£\, t^n e^{-\alpha t} = \frac{n!}{(s+\alpha)^{n+1}}$; so we identify $n = 2$ for the first term, $n = 1$ for the second term and $n = 0$ for the third term in expanded F(*s*) above.

Hence we can write $f(t)$ as $f(t) = t^2 e^{-2t} + 3te^{-2t} + 4e^{-2t} \quad t \geq 0$

Some representative MATLAB scripts for Laplace transform and inverse Laplace transform are given at the end of the chapter.

2.6 CONCEPT OF TRANSFER FUNCTION

The concept of transfer function is very important in analysis of control system. It is used to characterize the input-output relationships of components and subsystems and systems that can be described by linear, time-invariant, differential equations.

Definition 2.2 Transfer function

The *transfer function* of a linear, time-invariant system is defined as the ratio of the Laplace transform of the output to the Laplace transform of the input with all initial conditions in the system set equal to zero.

$$\text{Transfer function} = G(s) = \left.\frac{£\,[\text{output}]}{£\,[\text{input}]}\right|_{\text{zero initial conditions}} \tag{2.22}$$

Let us consider a third order linear time-invariant system described by the differential equation of the form:

$$a_0 \dddot{y} + a_1 \ddot{y} + a_2 \dot{y} + a_3 y = b_0 \dddot{x} + b_1 \ddot{x} + b_2 \dot{x} + b_3 x$$

where y is the system-output and x is the system-input and $\dot{y}$ stands for $\frac{dy}{dt}$. With the help of Theorem 2.5 we can take the Laplace transform of both the sides in the above equation to get:

$$a_0[s^3 Y(s) - s^2 y(0) - s\, y'(0) - y''(0)] + a_1[s^2 Y(s) - sy(0) - y'(0)]$$
$$+ a_2[sY(s) - y(0)] + a_3Y(s) = b_0[s^3 X(s) - s^2 x(0) - sx'(0) - x''(0)]$$
$$+ b_1[s^2 X(s) - sx(0) - x'(0)] + b_2[sX(s) - x(0)] + b_3X(s)$$

Now setting the initial values of $y(0), y'(0), y''(0), x(0), x'(0)$ and $x''(0)$ to zero we can write the transfer function from the definition as

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_0s^3 + b_1s^2 + b_2s + b_3}{a_0s^3 + a_1s^2 + a_2s + a_3}$$

In general, the transfer function of an nth order single-input, $x(t)$, single output, $y(t)$, linear time invariant system can be written in the form :

$$G(s) = \frac{Y(s)}{X(s)} = \frac{b_0s^m + b_1s^{m-1} + \ldots + b_{m-1}s + b_m}{a_0s^n + a_1s^{n-1} + \ldots + a_{n-1}s + a_n}, \text{ with } n \geq m \qquad (2.23)$$

The system dynamics can therefore, be represented by an algebraic equation in s domain by using the concept of transfer function.

Some observations about transfer function. The transfer function approach is extensively used in the analysis and design of control systems. It has already been pointed out that the concept of transfer function is applicable to linear, time-invariant systems that can be described by linear differential equations. In the block diagram analysis of systems, the transfer functions are written inside the blocks representing dynamics of the components and subsystems. Some observations concerning the transfer functions are noted below.

The transfer function of a given system *is unique and it is the property of the system itself and does not depend on the input and output and initial conditions*. The *concept is applicable to linear system only*. Nonlinear systems cannot be represented by transfer function concept.

The transfer function is a mathematical model, it does not give any information about the physical nature of the actual system. A third order transfer function may represent the dynamics of a field controlled D.C. motor controlling the position of a mechanical load or the voltage variation in an L-C-R circuit. Two entirely different physical systems may have similar structure of transfer functions.

Knowing the transfer function, one can study the response of the system when subjected to various inputs.

The transfer function of a system can be identified experimentally by using suitable inputs and noting the output of the system.

2.7 BLOCK DIAGRAMS

A control system normally consists of a number of components and subsystems. The dynamics of each component is represented by its transfer function and placed inside a box known as a block and their interrelationship, when they are non-interacting, is shown by connecting arrows, the direction of arrow indicating the direction of information or signal flow. Some rules are to be followed for simplifying the block diagram of a complex system, which are presented below.

Block diagrams. The block diagram of a system gives a perspective view of the functioning of a system, showing pictorially the interconnections among various components and subsystems and the cause and effect relationships by the direction of signal flow. It gives an overall picture of the functioning of the entire system which is not available from a purely abstract mathematical representation.

In the block diagram representation of a system, boxes known as *functional blocks* or simply *blocks* containing transfer functions of components or subsystems are inter connected by arrow lines based on cause and effect relationship of the physical variables of the control system. The arrows indicate the direction of signal flow, the signal can flow only in the direction of the arrow and the output from one block is input to the block connected by the arrow line. Thus in a block diagram representation of a control system, the inter dependence of the variables in different parts of the system are apparent at a single glance. The functional operation of the entire control system can be more readily comprehended from the inspection of block diagram than is possible from the physical system itself.

An element of the block diagram is shown in Fig. 2.4, where the arrowhead pointing toward the block indicates the input whereas the arrowhead moving away from the block represents the output.

Fig. 2.4 (a) Element of a block diagram (b) Summing point

The overall block diagram for the entire system can be obtained by connecting the blocks of the components and sub systems according to the signal flow and it is possible to evaluate the contribution of each component to the overall performance.

The block diagram does not carry any information on the physical nature or source of energy of the system but it contains information about system dynamics in the form of transfer function. As a consequence of this, many unrelated systems might have the same block diagram representation.

The block diagram of a given system is not unique and depending on the need, a number of different block diagrams can be drawn for the same system.

A block diagram consists of *blocks, summing points,* and *branch points* , which may be used to represent the functional operation of any linear control system. We have already mentioned that *blocks* are nothing but boxes containing transfer function of components and sub systems. The other two terms are explained below.

Summing point. The circle in Fig. 2.4(*b*) is a symbol used to represent the summing operation. The summing point performs algebraic summation of two or more signals with appropriate signs. In control systems, the signals need be added or subtracted depending on the control strategy. For instance, the feedback signal B(*s*) is to be subtracted from the reference signal R(*s*) to produce an error signal. This operation is represented by the summing point in part (*b*) of Fig. 2.4. Before the signals are connected to the summing point, one has to ensure that the signals are of the same dimension using proper transducers.

Branch point. A branch point on the block diagram is a point which gives rise to two or more signals to act as inputs to other blocks or summing points (vide Fig. 2.5).

Block diagram of a closed-loop system. A block diagram representation of a closed-loop system is shown in Fig. 2.5. The output Y(*s*) is fed back through a block H(*s*) to produce signal B(*s*), which is then subtracted at the summing point with the reference input R(*s*) to produce the error signal E(*s*). The output Y(*s*) is obtained by multiplying the error E(*s*) with transfer function G(*s*). The block diagram clearly shows the closed-loop nature of the system.

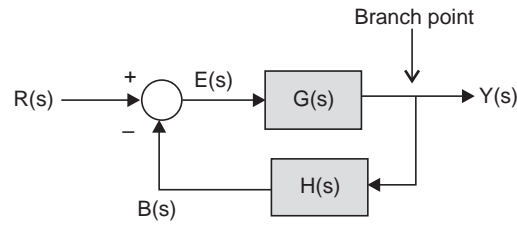



Fig. 2.5 Block diagram of a closed-loop system.

We have already mentioned that all the signals connected to the summing point should be of the same physical nature for proper summing operation. For instance, in a temperature control system, the output signal is the temperature. If the reference temperature is set by a potentiometer, it is represented by voltage. The output signal, which has the dimension of temperature, must be converted by a sensor like thermocouple to a voltage before it can be compared with the reference input. The thermocouple transfer function is represented by H(s), as shown in Fig. 2.5 and the feedback signal that is connected to the summing point for comparison with the input is B(s) = H(s)Y(s).

Loop-gain and closed-loop transfer function. Referring to Fig. 2.5, if the loop is opened after the feedback element H(s), the ratio of the feedback signal B(s) to the actuating error signal E(s) is called the loop-gain, which is given by:

$$\text{Loop Gain} = B(s)/E(s) = G(s)H(s)$$

The ratio of the output Y(s) to the actuating error signal E(s) is called the forward path transfer function, so that Forward path transfer function = Y(s)/E(s) = G(s)

Closed-loop transfer function. With reference to the block diagram of Fig. 2.5, let us find the relation between the system output Y(s) and system input R(s). Using the definition of transfer function we can write:

$$Y(s) = G(s)E(s)$$

$$E(s) = R(s) - B(s) = R(s) - H(s)Y(s)$$

Substituting the value of E(s) from the second equation to the first we get

$$Y(s) = G(s)[R(s) - H(s)Y(s)]$$

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{[1 + G(s)\,H(s)]} \tag{2.24}$$

The transfer function between the system output Y(s) and the system input R(s) in Fig. 2.5 is the *closed-loop transfer function*. The closed loop transfer function in Equation (2.24) is found to depend on the forward path transfer function G(s) and the loop gain G(s)H(s). From Equation (2.24), the output Y(s) may be written as :

$$Y(s) = \frac{G(s)\,R(s)}{[1 + G(s)\,H(s)]}$$

The above equation reveals that the output of the closed-loop system depends on the input as well as on the closed-loop transfer function.

Suppression of disturbance by a closed-loop system. Let us consider the closed loop system in Fig. 2.6 where the plant is subjected to a disturbance D(s). When the reference input R(s) and disturbance are both present in a linear system, their effect will be studied by the principle of superposition. The outputs corresponding to each input will be found

independent of the other and added to give the overall output. In order to examine the effect of the disturbance D(s) on the system output, we assume that the system is initially at rest with R(s) = 0, then we calculate the response $Y_D(s)$ to the disturbance only. The output response due to disturbance is found from

$$\frac{Y_D(s)}{D(s)} = \frac{G_2(s)}{1 + G_1(s)\,G_2(s)H(s)}$$

The response $Y_R(s)$ due to the reference input R(s) only is found from the following relation obtained from Fig. 2.6 with D(s) = 0

$$\frac{Y_R(s)}{R(s)} = \frac{G_1(s)\,G_2(s)}{1 + G_1(s)\,G_2(s)H(s)}$$

The output response when both the reference and the disturbance are present simultaneously, is obtained by adding the two responses found separately.

If the transfer functions $G_1(s)$, $G_2(s)$ and H(s) are such that they satisfy the conditions $| G_1(s)H(s) | >> 1$ and $| G_1(s)G_2(s)H(s) | >> 1$, then the closed-loop transfer function $Y_{D(s)}/D(s)$ will become very small, making the effect of the disturbance at the output negligibly small. This is one of the benefits of using a closed-loop system. We also notice that the closed-loop transfer function $Y_R(s)/R(s)$ becomes $1/H(s)$ when $| G_1(s)G_2(s)H(s) | >> 1$. Under this condition, the output $Y_R(s)$ due to the reference input R(s) depends entirely on H(s) and becomes independent of $G_1(s)$ and $G_2(s)$. This implies that the closed-loop transfer function $Y_R(s)/R(s)$ remains unaffected by any variations of $G_1(s)$ and $G_2(s)$ so long the condition $| G_1(s)G_2(s)H(s) | >> 1$ is satisfied. This is very significant, since the feedback path transfer function H(s) can be made insensitive to changes in operating conditions by keeping it in controlled environment. Also with H(s) = 1, Y(s) can be made equal to R(s).

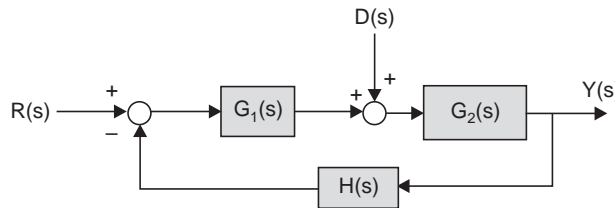



Fig. 2.6 Closed loop system subjected to disturbance D(s)

2.7.1 Block Diagram Reduction

In the block diagram representation, the signal from the output of one block is used as input to another block. The transfer functions of the two cascaded blocks may be replaced by a single transfer function which is equal to their product contained in a single block provided the signal output of the first block is not loaded in the physical system by the presence of the physical component represented by the second block. In case of loading of the first component by the presence of second, the transfer function of the two components taken together should be derived before representing them by a single block.

Under this assumption, any number of cascaded blocks can be replaced by a single block containing the transfer function which is the product of transfer functions of the individual blocks.

Remembering the above facts, a few rules may be framed for simplification of complex block diagram containing many feedback loops. Subsequent mathematical analysis becomes

simpler with reduced block diagram. A few of these rules are presented in Table 2.2. These rules are obtained by writing the relation between the variables involved in alternative forms. Sometimes, the transfer functions in new blocks become more complex because of the fact that new poles and new zeros are generated in the equivalent block diagram. However, one should keep the following in mind when simplifying a block diagram :

1. The product of the transfer functions in the forward direction must remain unchanged.
2. The product of the transfer functions around the loop must also remain unchanged.

Example 2.5 Let us simplify the block diagram given in Fig. 2.7(*a*) by using the rules presented in Table 2.2.

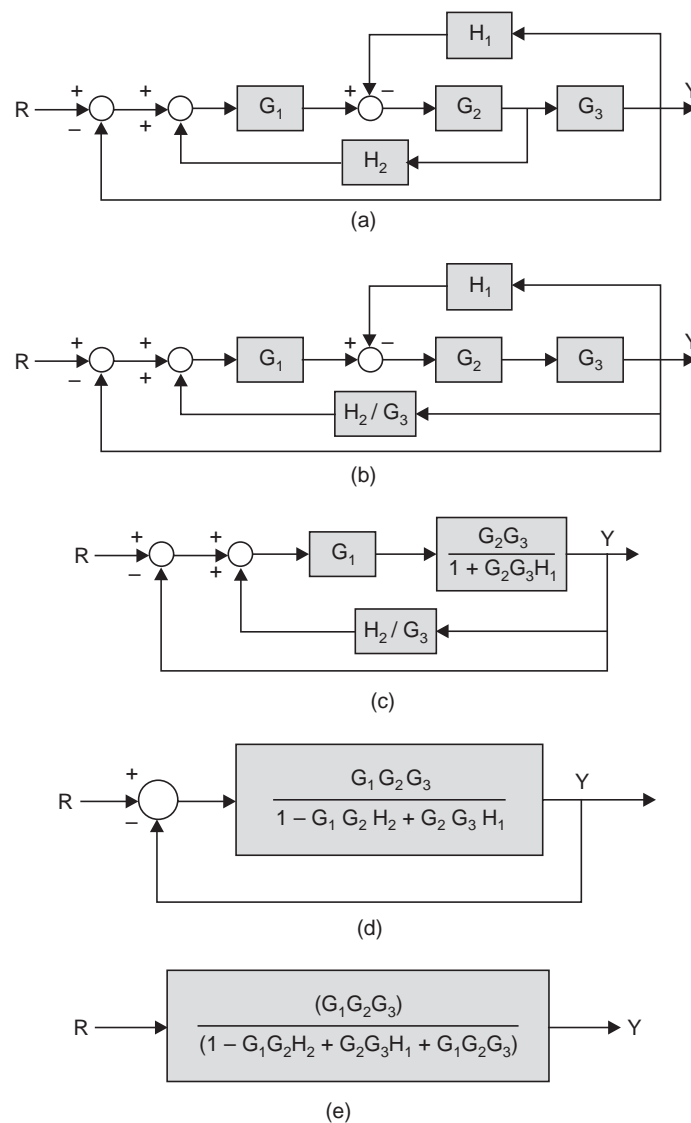
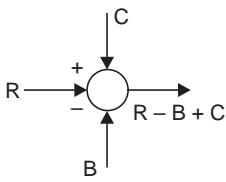
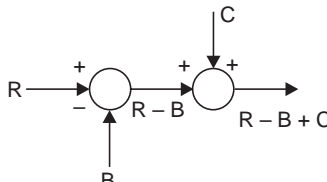
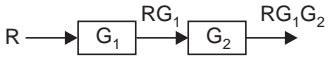
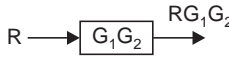
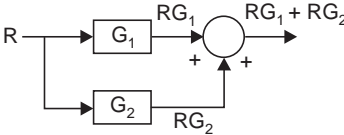
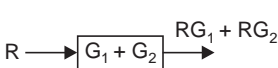
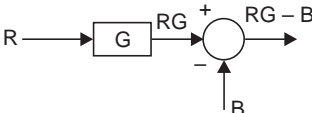
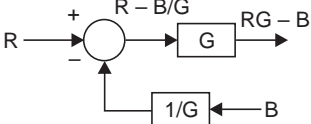
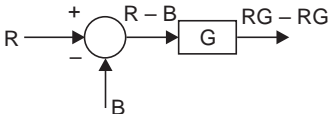
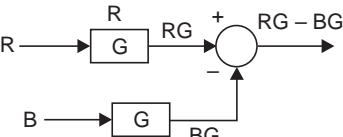
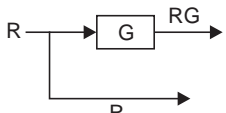
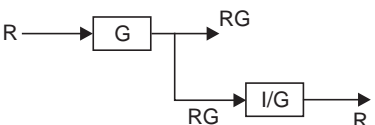
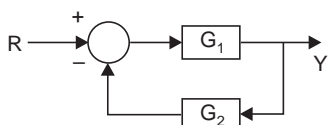
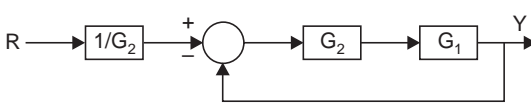



Fig. 2.7 (a) Block Diagram of a system; (b)–(e) Steps for Block diagram reduction

We move the branching point of the negative feedback loop containing H_2 to the output point as shown in Fig. 2.7 (*b*). We eliminate the negative feedback loop containing H_1, to get the Fig. 2.7(*c*). Then we eliminate the loop containing H_2/G_3 to get Fig. 2.7(*d*). Finally, eliminating the unity feedback loop results in Fig. 2.7(*e*).

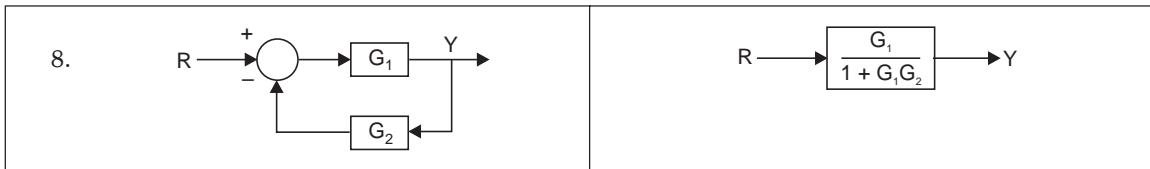
The closed-loop transfer function $Y(s)/R(s)$ is, therefore, given by

$$\frac{Y}{R} = \frac{G_1G_2G_3}{1 + G_1G_2H_1 + G_2G_3H_1 + G_1G_2G_3} \tag{2.25}$$

Table 2.2. Rules for Block Diagram Reduction

	Original Block Diagram	*Equivalent Block Diagram*
1.	C, +, R, −, R – B + C, B	C, +, +, R, +, −, R – B, R – B + C, B
2.	R, G_1, RG_1, G_2, RG_1G_2	R, G_1G_2, RG_1G_2
3.	R, G_1, RG_1, +, +, $RG_1 + RG_2$, G_2, RG_2	R, $G_1 + G_2$, $RG_1 + RG_2$
4.	R, G, RG, +, −, RG – B, B	R, +, −, R – B/G, G, RG – B, 1/G, B
5.	R, +, −, R – B, B, G, RG – RG	R, G, R, RG, +, −, RG – BG, B, G, BG
6.	R, G, RG, R	R, G, RG, RG, I/G, R
7.	R, +, −, G_1, Y, G_2	R, $1/G_2$, +, −, G_2, G_1, Y

...(Contd.)

8.

2.8 SIGNAL FLOW GRAPH REPRESENTATION

An alternate approach to block diagram representation of control system dynamics, is the signal flow graph approach, introduced by S.J. Mason [12–13]. However, both the signal flow graph approach and the block diagram approach provide the same information for system analysis and one is not superior to the other in any sense of the term.

2.8.1 Signal Flow Graphs

A signal flow graph is used to represent graphically a set of simultaneous linear algebraic equations. After transforming linear differential equations into algebraic equations in complex variable *s* the signal flow graph method may be employed for analysis of control systems.

The variables in the algebraic equation are represented by nodes and a graph is formed by connecting the nodes with directed branches in such a way as to satisfy the algebraic equations. The signal can flow only in the direction of the arrow of the branch and it is multiplied by a factor indicated along the branch, which happens to be the coefficient of the algebraic equation. The signal flow graph depicts the flow of signals from one point of a system to another in a cause and effect relationships- the variable at the arrow head being the dependent variable.

As an illustration, consider the algebraic equation $x_2 = g_{12}x_1$, where x_1 is the independent variable and x_2 is the dependent variable. The equation may be represented as a signal flow graph as shown in Fig. 2.8, where signal can flow from x_1 to x_2 only in the direction of the arrow not the reverse. The magnitude of the signal at x_2 is obtained by multiplying the signal x_1 with branch gain g_{12}.

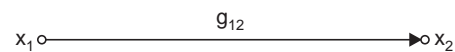



Fig. 2.8 A simple signal flow graph

A gain formula, known as Mason's gain formula, may be used to obtain the relationships among variables of a control system represented by a signal flow graph. It does not need any reduction of the original graph, as was the case with the block diagram approach.

Terminologies. A few terminologies are to be introduced before we can discuss signal flow graphs.

Node: A node is a point representing a variable or signal, (*e.g.*, x_1, x_2 in Fig. 2.8).

Transmittance: The transmittance is a gain, which may be real or complex between two nodes (g_{12} in Fig. 2.8).

Branch: A branch is a directed line segment between two nodes. The transmittance is the gain of a branch.

Input node: An input node has only outgoing branches and this represents an independent variable (*e.g.*, x_1 in Fig. 2.8).

Output node: An output node has only incoming branches representing a dependent variable (*e.g.*, x_2 in Fig. 2.8). In order to comply with this definition an additional node may be introduced with unity gain to act as an output node (*e.g.*, x_3 in Fig. 2.9).

Mixed node: A mixed node is a node that has both incoming and outgoing branches (*e.g.*, x_2, x_3 in Fig. 2.9).

Path: Any continuous unidirectional succession of branches traversed in the indicated branch direction is called a path.

Loop: A loop is a closed path (*e.g.*, the path $x_2\, x_3\, x_2$ in Fig. 2.9).

Loop gain: The loop gain is the product of the branch transmittances of a loop ($g_{23}\, g_{32}$ in Fig. 2.9).

Nontouching loops: Loops are non-touching if they do not have any common nodes.

Forward path: A forward path is a path from an input node to an output node along which no node is encountered more than once (*e.g.*, $x_1x_2x_3$ and $x_4\, x_3$ in Fig. 2.9)

Feedback path (loop): A path which originates and terminates on the same node along which no node is encountered more than once is called a feedback path (*e.g.*, the path $x_2\, x_3\, x_2$ in Fig. 2.9).

Path gain: The product of the branch gains encountered in traversing the path is called the path gain (*e.g.*, $g_{12}g_{23}$ for forward path $x_1x_2x_3$ and g_{43} for forward path x_4x_3 in Fig. 2.9).

Loop gain: The product of the branch gains of the branches forming that loop is called loop gain (*e.g.*, $g_{23}\, g_{32}$ in Fig. 2.9).

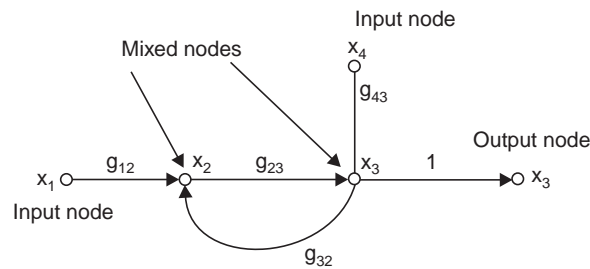



Fig. 2.9 Signal flow graph

2.8.2 Properties of Signal Flow Graphs

A few important properties of signal flow graphs are noted below:

1. A branch indicates the functional dependence of one variable on another.
2. A node performs summing operation on all the incoming signals and transmits this sum to all outgoing branches.

2.8.3 Signal Flow Graph Algebra

With the help of the foregoing terminologies and properties, we can draw the signal flow graph of a linear system. For convenience, the input nodes are placed to the left and the output nodes to the right in the signal flow graph. The independent variables of the equations become the input nodes while the dependent variables become the output nodes in the graph. As already mentioned, the coefficients of the algebraic equations are written as the branch transmittances.

The Mason's gain formula which will be presented shortly, is used to find the relationship between any variable considered as input and any other variable taken as output. For applying the Mason's gain formula, we use the following rules:

1. The value of an output node with one incoming branch, as shown in Fig. 2.10(*a*) is found by multiplying the input variable with branch transmittance, $x_2 = g_{12}x_1$.

2. The total transmittance of cascaded branches is equal to the gain of the path formed with all the branches. Therefore, the cascaded branches can be replaced by a single branch with transmittance equal to the path gains, as shown in Fig. 2.10(*b*).
3. Parallel branches may be replaced by a single branch with transmittance which is equal to the sum of the transmittances, as shown in Fig. 2.10(*c*).
4. A mixed node may be eliminated, by modifying the graph as shown in Fig. 2.10(*d*).
5. A loop may be eliminated by modifying the graph, as shown in Fig. 2.10(*e*).

We find in Fig. 2.10(*e*) that $x_3 = g_{23}\, x_2$, and $x_2 = g_{12}\, x_1 - g_{32}\, x_3$. Hence

$$x_3 = g_{23}\, g_{12}\, x_1 - g_{23}\, g_{32}\, x_3$$

or

$$x_3 = \frac{g_{12} g_{23}}{1 + g_{23} g_{32}}\, x_1 \tag{2.26}$$

(*a*) x1 —g12→ x2

(*b*) x1 —g12→ x2 —g23→ x3 ⟹ x1 —g12 g23→ x3

(*c*)

(*d*)

(*e*)

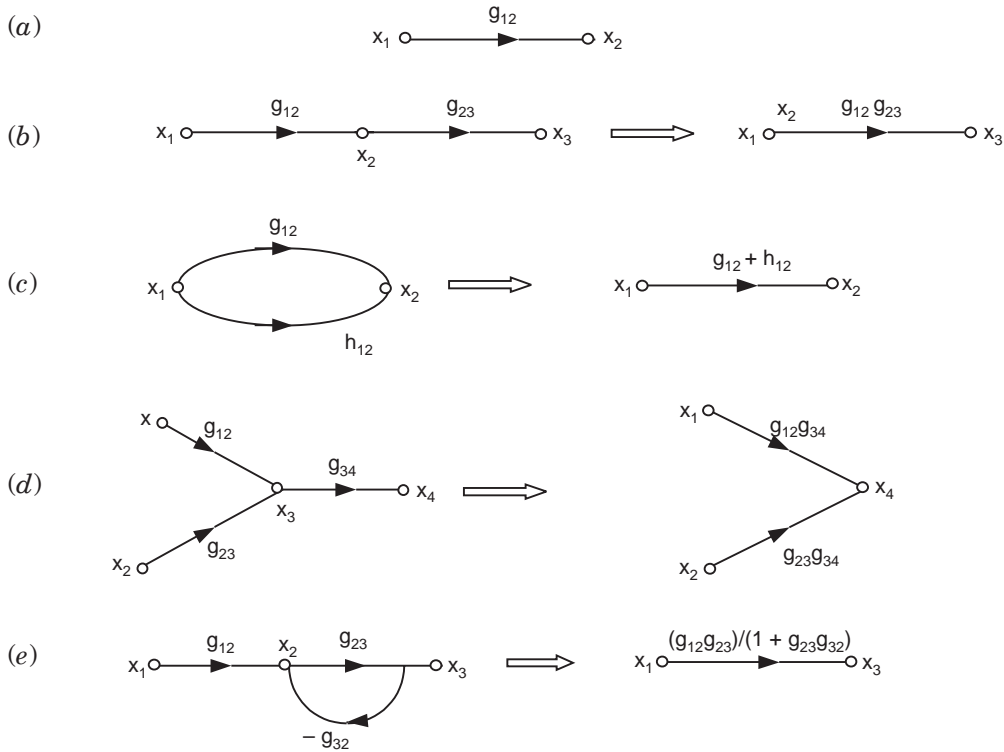



Fig. 2.10 Signal flow graph and simplification

2.8.4 Representation of Linear Systems by Signal Flow Graph

Signal flow graphs can be used conveniently in linear-system analysis, after the graph has been drawn from the system equations. The foregoing rules may be routinely used to find the relation between an input and output variable.

Let us consider a system with u_1 and u_2 as input variables and x_1 and x_2, as output variables described by the following set of equations:

$$x_1 = a_{11}x_1 + a_{12}x_2 + b_1u_1 \tag{2.27}$$

$$x_2 = a_{21}x_1 + a_{22}x_2 + b_2u_2 \tag{2.28}$$

A graphical representation of these two simultaneous equations, known as signal flow graph for the system are obtained as follows: First we locate the nodes x_1, x_2, u_1 and u_2 as shown in Fig. 2.11(a). We note that the coefficient a_{ij} is the transmittance between the nodes x_i and x_j.

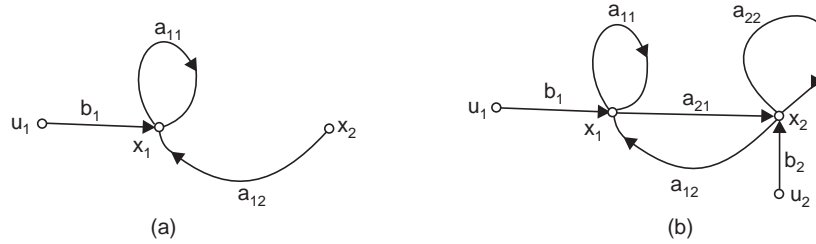



Fig. 2.11 Signal flow graphs representing (a) Equation (2.27); (b) complete signal flow graph for the system described by Equations (2.27) and (2.28)

Equation (2.27) indicates that x_1 is equal to the sum of the three signals $a_{11}x_1$, $a_{12}x_2$, and b_1u_1. The signal flow graph representing Equation (2.27) is shown in Fig. 2.11(a). Equation (2.28) indicates that x_2 is equal to the sum of $a_{21}x_1$, $a_{22}x_2$, and b_2u_2. The complete signal flow graph for the given simultaneous equations is shown in Fig. 2.11(b).

Fig. 2.12 shows the signal flow diagram representation of transfer function blocks. While drawing the signal flow graph from the block diagram, the sign associated with a signal be written with the transmittance. The negative sign of feedback path signal in Fig. 2.12(c) is included with the transmittance H(s) in Fig. 2.12 (d).

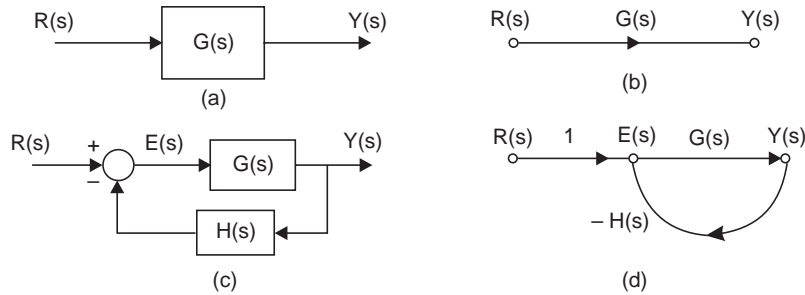



Fig. 2.12 Block diagrams and corresponding signal flow graphs

The overall gain from an input to an output may be obtained directly from the signal flow graph by inspection, by use of Mason's formula, or by a reduction of the graph to a simpler form.

2.8.5 Mason's Gain Formula

In a control system the transfer function between any input and any output may be found by Mason's Gain formula. Mason's gain formula is given by

$$M = \frac{1}{\Delta} \sum_k M_k \Delta_k \tag{2.29}$$

where M_k = path gain of kth forward path

Δ = determinant of the graph

= 1 – (sum of all individual loop gains) + (sum of gain products of all possible combinations of two non-touching loops) – (sum of gain products of all possible combinations of three non-touching loops) + ……

$$= 1 - \sum_i L_i + \sum_{i,j} L_i L_j - \sum_{i,j,k} L_i L_j L_k + \ldots\ldots$$

where $\sum_i L_i$ = sum of all individual loop gains in the graph

$\sum_{i,j} L_i L_j$ = sum of gain products of all possible combinations of two non-touching loops of the graph

$\sum_{i,j,k} L_i L_j L_k$ = sum of gain products of all possible combinations of three non- touching loops

and Δ_k = the value of Δ computed by removing the loops that touch the kth forward path

We shall now illustrate the use of Mason's gain formula by means of examples.

Example 2.6 Let us consider the block diagram of a system shown in Fig. 2.13. The signal flow graph for this system may be drawn with nodes $X_1(s)$, $X_2(s)$, $X_3(s)$, $X_4(s)$, $X_5(s)$ and $Y_1(s)$, as shown in Fig. 2.14. We are interested to obtain the closed-loop transfer function Y(*s*)/R(s) by use of Mason's gain formula.

In this system there are two forward paths between the input R(s) and the output Y(s). The forward path gains are

$M_1 = G_4(s)$ for the path $R(s)E(s)X_2(s)Y(s)$

$M_2 = G_1(s)G_2(s)G_3(s)$ for the path $R(s)E(s)X_4(s)X_3(s)X_1(s)Y(s)$

From Fig. 2.14, we observe that there are five individual loops. The gains of these loops are

$L_1 = -G_1(s)G_2(s)H_2(s)$ for the loop $E(s)X_4(s)X_3(s)E(s)$

$L_2 = -G_1(s)G_2(s)G_3(s)$ for the loop $E(s)X_4(s)X_3(s)X_1(s)Y(s)E(s)$

$L_3 = -G_4$ for the loop $E(s)X_2(s)Y(s)E(s)$

$L_4 = -G_2(s)G_3(s)H_1(s)$ for the loop $X_4(s)X_3(s)X_1(s)Y(s)X_4(s)$

$L_5 = G_4(s)H_1(s)G_2(s)H_2(s)$ for the loop $E(s)X_2(s)Y(s)X_4(s)X_3(s)E(s)$

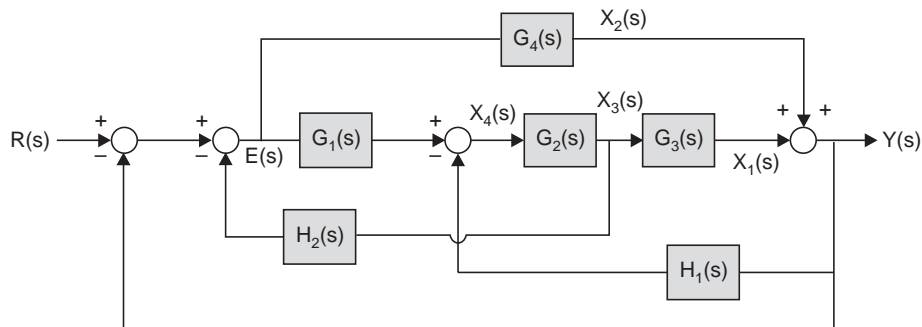



Fig. 2.13 Block Diagram representation of control system

We observe that there are no non-touching loops in this signal flow graph. So, the determinant Δ for the graph is given by

$$\Delta = 1 - (L_1 + L_2 + L_3 + L_4 + L_5)$$
$$= 1 + G_1(s)G_2(s)H_2(s) + G_2(s)G_3(s)H_1(s) + G_1(s)G_2(s)G_3(s)$$
$$+ G_4(s) - G_4(s)H_1(s)G_2(s)H_2(s)$$

The value of Δ_1 is computed in the same way as Δ by removing the loops that touch the first forward path M_1. In this example, since path M_1 touches all the five loops, Δ_1 is found as :

$$\Delta_1 = 1$$

Proceeding in the same way, we find $\Delta_2 = 1$, since all the five loops also touch the second forward path.

Therefore, the closed loop transfer function between the input R(*s*) and the output Y(*s*), is given by

$$\frac{C(s)}{R(s)} = M = \frac{M_1\Delta_1 + M_2\Delta_2}{\Delta}$$

$$= \frac{G_4(s) + G_1(s)G_2(s)G_3(s)}{1 + G_1(s)G_2(s)H_2(s) + G_2(s)G_3(s)H_1(s) + G_1(s)G_2(s)G_3(s) + G_4(s) - G_2(s)G_4(s)H_1(s)H_2(s)}$$

Mason's gain formula thus helps us to find the ratio Y(*s*)/R(*s*) without any reduction of the graph in Fig. 2.14, even though, the same result could have been obtained by reduction of the blocks in the Fig. 2.13.

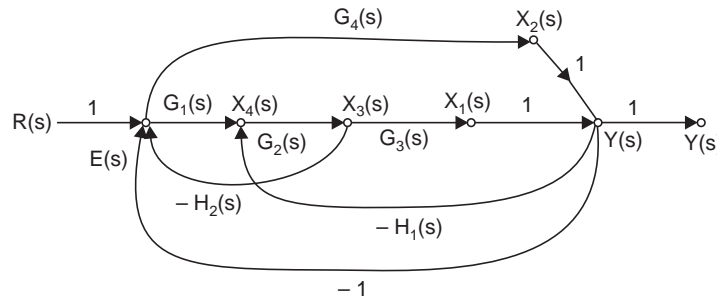



Fig. 2.14 Signal flow graph for the system shown in Fig. 2.13

Example 2.7 Let us consider the system shown in Fig. 2.15 and use the Mason's gain formula to obtain the closed-loop transfer function Y(*s*)/R(*s*).

From the Fig., we observe that there are three forward paths between the input R(*s*) and the output Y(*s*). The respective forward path gains are found to be :

$M_1 = G_1G_2G_4G_6G_7$ for the path R(*s*) $x_4\ x_3\ x_2\ x_1$ Y(*s*)

$M_2 = G_1G_3G_6G_7$ for the path R(*s*) $x_4\ x_2\ x_1$ Y(*s*)

$M_3 = G_1G_2G_5$ for the path R(*s*) $x_4\ x_3$ Y(*s*)

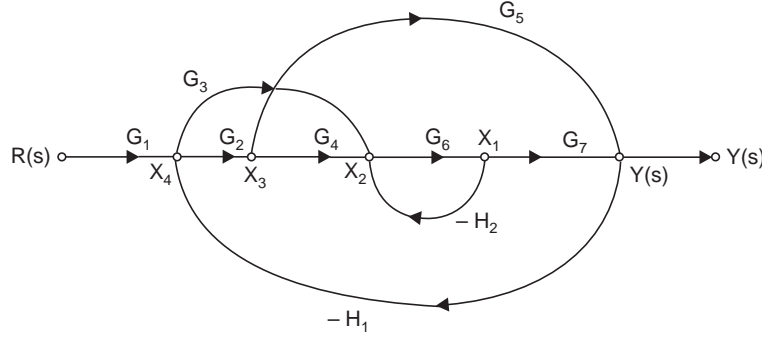



Fig. 2.15 Signal flow graph for a system

There are four individual loops with the following gains:

$L_1 = -G_6H_2$ for the loop $x_2x_1x_2$

$L_2 = -\ G_2G_5H_1$ for the loop $x_4x_3Y(s)x_4$

$L_3 = -\ G_3G_6G_7H_1$ for the loop $x_4x_2x_1Y(s)x_4$

$L_4 = -\ G_2G_4G_6G_7H_1$ for the loop $x_4x_3\ x_2x_1Y(s)x_4$

Since the loops L_1 and L_2 are the only not touching loops in the graph, the determinant Δ will be given by

$$\Delta = 1 - (L_1 + L_2 + L_3 + L_4) + L_1L_2 \qquad (2.30)$$

The value of Δ_1 is obtained from Δ by removing the loops that touch first forward path M_1. So, by removing L_1, L_2, L_3, L_4, and L_1 L_2 from Equation (2.30), we get

$$\Delta_1 = 1$$

Similarly, Δ_2 is found as $\Delta_2 = 1$

We observe that L_1 is the only loop that does not touch the third forward path M_3 . Hence, by removing L_2, L_3, L_4, and L_1 L_2 from Equation (2.30), we get Δ_3 as

$$\Delta_3 = 1 - L_1 = 1 + G_6H_2$$

The closed-loop transfer function $Y(s)/R(s)$ is then given by

$$\frac{Y(s)}{R(s)} = M = \frac{1}{\Delta}\,(M_1\Delta_1 + M_2\Delta_2 + M_3\Delta_3)$$

$$= \frac{G_1G_2G_4G_6G_7 + G_1G_3G_6G_7 + G_1G_2G_5(1+G_6H_2)}{1+G_6H_2+G_2G_5H_1+G_3G_6G_7H_1+G_2G_4G_6G_7H_1+G_6H_2G_2G_5H_1}$$

Mason's gain formula may be used to establish the relationship between any input and any output. It is useful in reducing large and complex system diagrams in one step, without requiring step-by-step reductions. However, one must be very careful not to omit any loops and their combinations in calculating Δ and Δ_k.

A few more examples of the application of signal flow graphs will be found in Section 3.5.1 where the resolvent matrix is computed with Mason's gain formula that avoids matrix inversion.

2.9 VECTORS AND MATRICES

A convenient tool for state variable representation is matrix notation, so a brief discussion of matrix notation is in order.

A matrix is a rectangular array of elements arranged in m rows and n columns, which obeys a certain set of manipulations outlined below. The elements of the matrix may be real or complex numbers of variables of either time or frequency. The matrix A is represented as

$$A = [a_{ij}] = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \quad (2.31)$$

A matrix is classified by the number of its rows and columns: a matrix of m rows and n columns is a matrix of order $m \times n$, or an $m \times n$ matrix. If $m = n$, the matrix is called a square matrix. A matrix with n rows and only one column is called a column matrix or column vector. The notation for the column vector is a lowercase, boldface letter, as in

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \text{col}\,(x_1\; x_2\; x_3 \dots x_n) \quad (2.32)$$

Note that the elements of a vector have only one subscript, to indicate their location in the column. A matrix with one row and n columns is called a row matrix or row vector. The transpose of any matrix is formed by interchanging rows and columns of the matrix and is indicated by a prime (*i.e.,* A′) so that

$$A' = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ a_{12} & a_{22} & \dots & a_{m2} \\ \dots & \dots & \dots & \dots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \quad (2.33)$$

Designating a column matrix to be a vector is not precisely correct since a vector has a fixed geometric meaning while the column matrix is simply a representation of a given vector in one coordinate system. However, it is retained here since it is commonly used in the literature. A row matrix may be thought of as the transpose of a column matrix so that $\mathbf{x}'$ is the row matrix

$$\mathbf{x}' = [x_1\; x_2\; x_3 \dots x_n]$$

The determinant of the matrix $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ is written as $|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix}$ and is computed as $|A| = a_{11} \cdot a_{22} - a_{21} \cdot a_{12}$

Similarly the determinant of a 3 × 3 matrix $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$ (2.34)

is given by $|A| = a_{11} \cdot a_{22} \cdot a_{33} - a_{11} \cdot a_{23} \cdot a_{32} - a_{21} \cdot a_{12} \cdot a_{33} + a_{31} \cdot a_{12} \cdot a_{23}$

$$+ a_{21} \cdot a_{13} \cdot a_{32} - a_{31} \cdot a_{13} \cdot a_{22} \quad (2.35)$$

2.9.1 Minors, Cofactors and Adjoint of a Matrix

The minor of an element in a matrix is the determinant obtained by deleting the row and the column corresponding to that element. For instance, in the 3 × 3 matrix A in (2.34), the minor corresponding to the element a_{11} is given by:

Minor of $a_{11} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}$; similarly minor of $a_{12} = \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix}$ and minor of $a_{32} = \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix}$;

The cofactor of an element in a matrix is its minor with appropriate sign. The sign of the cofactor corresponding to an element at the ith row and jth column of a matrix is $(-1)^{i+j}$. Denoting the cofactor of an element by the corresponding capital letter, we have

$$A_{11} = (-1)^{1+1} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} = \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix}; \text{ similarly } A_{12} = (-1)^{1+2} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} = -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix}$$

$$A_{32} = (-1)^{3+2} \begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} = -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \tag{2.36}$$

The determinant of a matrix can be expanded along a row or column in terms of its cofactors. Expanding along a row, the determinant of the 3 × 3 matrix A in (2.34) may be written as:

$$|\,A\,| = a_{11}A_{11} + a_{12}A_{12} + a_{13}A_{13} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} \tag{2.37}$$

$$= a_{11} \cdot a_{22} \cdot a_{33} - a_{11} \cdot a_{23} \cdot a_{32} - a_{21} \cdot a_{12} \cdot a_{33} + a_{31} \cdot a_{12} \cdot a_{23} + a_{21} \cdot a_{13} \cdot a_{32} - a_{31} \cdot a_{13} \cdot a_{22}$$

which is the same result as in (2.35)

If we form a matrix with all the cofactors of the A matrix in (2.34) we get

$$\text{Cofactor matrix of A} = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}$$

The transpose of this cofactor matrix is known as Adjoint matrix and written as Adj A. The adjoint of the A matrix in (2.34) is given by :

$$\text{Adj A} = \begin{bmatrix} A_{11} & A_{21} & A_{31} \\ A_{12} & A_{22} & A_{32} \\ A_{13} & A_{23} & A_{33} \end{bmatrix} \tag{2.38}$$

Matrices are also classified by their rank [14]. The rank of a matrix is the largest number r such that at least one $r \times r$ minor of the matrix is nonzero and every minor of order $r + 1$ is zero. It is obvious that the rank of a matrix may never be larger than the smaller dimension of a matrix, that is $r \le$ minimum (m, n). Consider, for example, the matrix

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 1 & 3 & 4 \\ 3 & 6 & 9 \end{bmatrix}$$

In this case, the single third-order determinant of the matrix itself is zero. However, every second-order minor is nonzero, and therefore the matrix has rank 2.

Another type of classification of square matrices is concerned with the elements of the matrix. If $a_{ij} = a_{ji}$, for all i and j, the matrix is called symmetric, and we see then A′ = A. For example, the matrix.

$$A = \begin{bmatrix} 2 & 3 & 5 \\ 3 & 6 & 8 \\ 5 & 8 & 9 \end{bmatrix} \text{ is symmetric.}$$

If A is square, and if $a_{ij} = 0$, for $i \neq j$, then A is called a diagonal matrix. In the special case when A is diagonal and $a_{ii} = 1$ for all i, the matrix is known as the identity matrix, designated as

$$I = \begin{bmatrix} 1 & 0 & 0 & . & . & 0 & 0 \\ 0 & 1 & 0 & . & . & 0 & 0 \\ 0 & 0 & 1 & . & . & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & 1 & 0 \\ 0 & 0 & 0 & . & . & 0 & 1 \end{bmatrix} \qquad (2.39)$$

2.10 INVERSION OF A NONSINGULAR MATRIX

A square matrix A is called a nonsingular matrix if there exists a matrix B such that AB = BA = I, where I is the identity matrix with the same order as that of A and B. Such a B matrix is denoted by A^{-1} and is called the inverse of A. The inverse of A exists if the determinant | A | is nonzero. When the determinant | A | is zero, the inverse of A does not exist and it is called a singular matrix. If P and Q are nonsingular matrices , then the product PQ is also a nonsingular matrix and it will satisfy the following:

$$(PQ)^{-1} = Q^{-1}P^{-1}$$

The inverse of the transpose matrix satisfies the condition $(P')^{-1} = (P^{-1})'$

(*a*) **Some useful properties and relation of inverse matrices**

(*i*) If $\alpha \neq 0$ is scalar and A is $n \times n$ nonsingular matrix, then $(\alpha A)^{-1} = \frac{1}{\alpha} A^{-1}$

(*ii*) The determinant of A^{-1} is the reciprocal of the determinant of A, that is,

$$| A^{-1} | = \frac{1}{|A|}$$

(*iii*) For a 2 × 2 nonsingular matrix A given by $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$, the inverse is found in terms of its adjoint and determinant as $A^{-1} = \frac{\text{Adj A}}{\Delta} = \frac{1}{\Delta}\begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$ or $\mathbf{A}^{-1} = \frac{1}{\Delta}\begin{bmatrix} \mathbf{a}_{22} & -\mathbf{a}_{21} \\ -\mathbf{a}_{12} & \mathbf{a}_{11} \end{bmatrix}$; where $\Delta = a_{11}a_{22} - a_{12}a_{21} \neq 0$ is the determinant of the matrix A.

(*iv*) For a 3 × 3 nonsingular matrix A given by

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

the inverse is computed in terms of its adjoint and determinant as

$$A^{-1} = \frac{\text{Adj. A}}{\Delta}$$

$$A^{-1} = \frac{1}{\Delta}\begin{bmatrix} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{12} & a_{13} \\ a_{32} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{12} & a_{13} \\ a_{22} & a_{23} \end{vmatrix} \\ -\begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{13} \\ a_{31} & a_{33} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{13} \\ a_{21} & a_{23} \end{vmatrix} \\ \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix} & -\begin{vmatrix} a_{11} & a_{12} \\ a_{31} & a_{32} \end{vmatrix} & \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \end{bmatrix}$$

where $\Delta \neq 0$ is the determinant of the matrix A given by

$$|\,A\,| = a_{11} \,.\, a_{22} \,.\, a_{33} - a_{11} \,.\, a_{23} \,.\, a_{32} - a_{21} \,.\, a_{12} \,.\, a_{33} + a_{31} \,.\, a_{12} \,.\, a_{23} + a_{21} \,.\, a_{13} \,.\, a_{32} - a_{31} \,.\, a_{13} \,.\, a_{22}$$

In MATLAB command window, after entering the elements of the matrix A, *inv(A)* gives the inverse of the matrix. The 2 × 2 matrix A is entered as

$>> A = [a_{11}\ a_{12};\ a_{21}\ a_{22}];$

and the inverse of nonsingular matrix A assigned as B is obtained by the command:

>> *B* = *inv*(A) and determinant of A is found as >> *delta A* = *det* (A)

2.11 EIGEN VALUES AND EIGEN VECTORS

An n-dimensional vector $\mathbf{x}$ with components as in relation (2.32) may be transformed to another n-dimensional vector $\mathbf{y}$ by a $n \times n$ matrix A of the form given in relation (2.31) by a linear transformation

$$\mathbf{y} = A\mathbf{x} \tag{2.40}$$

Very often it is required to transform a vector to itself or it's scalar multiple. In such cases, for some scalar λ, the Equation (2.40) may be written as

$$\lambda\,\mathbf{x} = A\mathbf{x} \tag{2.41}$$

The above equation may be rearranged as

$$(\lambda I - A)\,\mathbf{x} = 0 \tag{2.42}$$

which represents n homogeneous equations as shown below:

$$\begin{aligned} (a_{11} - \lambda)\, x_1 + a_{12}\, x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n &= 0 \\ a_{21}x_1 + (a_{22} - \lambda)\, x_2 + a_{23}x_3 + \ldots\ldots + a_{2n}x_n &= 0 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots& \\ a_{n1}x_1 + a_{n2}\, x_2 + a_{n3}x_3 + \ldots\ldots + (a_{nn} - \lambda)\, x_n &= 0 \end{aligned} \tag{2.43}$$

The homogeneous equation (2.43) will have non-trivial solution if the coefficient matrix satisfies the condition

$$\det(\lambda I - A) = |\,\lambda I - A\,| = 0 \tag{2.44}$$

The Equation (2.44) is known as the characteristic equation of transformation, which is also the characteristic equation of the matrix A. The characteristic equation has n roots λ_i , $i = 1$ to n, known as Eigen values and corresponding to each root, Equation (2.42) or (2.43) has nonzero solution $\mathbf{x}$ which are known as Eigen Vectors.

Example 2.8 Let us consider the matrix A given by

$$A = \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix}$$

so that $(\lambda I - A)$ becomes $\begin{bmatrix} \lambda & 0 \\ 0 & \lambda \end{bmatrix} - \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix} = \begin{bmatrix} \lambda & -1 \\ 6 & \lambda+5 \end{bmatrix}$ (2.45)

The characteristic equation (2.45) is found to be $|\lambda I - A| = (\lambda + 2)(\lambda + 3) = 0$

Two eigen values are found from the characteristic equation as $\lambda_1 = -2$ and $\lambda_2 = -3$

The first eigen vector $\mathbf{x}_1$ is found from Equation (2.42) with $\lambda_1 = -2$. Using (2.43) we have

$$\begin{bmatrix} -2 & -1 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Both the equations are satisfied with $x_{12} = -2x_{11}$.

Therefore, the eigen vector corresponding to the first eigen value $\lambda_1 = -2$ is given by

$$\mathbf{x}_1 = \begin{bmatrix} x_{11} \\ x_{12} \end{bmatrix} = \begin{bmatrix} -k \\ 2k \end{bmatrix},$$

where k is an arbitrary scale factor. By choosing the scale factor such that the largest component of the eigen vector is of unity magnitude, the scaled eigen vector is

$$\mathbf{x}_1 = \begin{bmatrix} -1/2 \\ 1 \end{bmatrix}$$

Similarly, for second eigen value $\lambda_1 = -3$, the scaled eigen vector is found to be

$$\mathbf{x}_1 = \begin{bmatrix} -1/3 \\ 1 \end{bmatrix}$$

It is to be noted that the eigen values and eigen vectors may be complex, even though the elements of the matrix A are real. When the eigen values are complex, they occur in conjugate pairs, since the coefficients of the characteristic equation are real.

2.12 SIMILARITY TRANSFORMATION

Two matrices A and B of dimension $n \times n$ are said to be similar, if there exists a nonsingular matrix P such that

$$B = P^{-1}AP \tag{2.46}$$

The matrix B is said to be obtained by *similarity transformation* and P is the transformation matrix.

2.12.1 Diagonalization of Matrices

A similarity transformation of particular interest to us is the case when the matrix B in (2.46) is a diagonal matrix of the form :

$$B = \begin{bmatrix} \lambda_1 & 0 & 0 & . & . & 0 & 0 \\ 0 & \lambda_2 & 0 & . & . & 0 & 0 \\ 0 & 0 & \lambda_3 & . & . & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & \lambda_{n-1} & 0 \\ 0 & 0 & 0 & . & . & 0 & \lambda_n \end{bmatrix} \tag{2.47}$$

Chapter 2

Here the eigen values $\lambda_{i,}$ $i = 1 \ldots n$ are assumed to be distinct.

If we write the transformation matrix P in terms of its columns P = [$\mathbf{p}_1\ \mathbf{p}_3\ \mathbf{p}_3 \ldots \mathbf{p}_n$], then the Equation (2.46) can be written as:

$$[\ \mathbf{p}_1\ \mathbf{p}_3\ \mathbf{p}_3 \ldots \mathbf{p}_n] \begin{bmatrix} \lambda_1 & 0 & 0 & . & . & 0 & 0 \\ 0 & \lambda_2 & 0 & . & . & 0 & 0 \\ 0 & 0 & \lambda_3 & . & . & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & \lambda_{n-1} & 0 \\ 0 & 0 & 0 & . & . & 0 & \lambda_n \end{bmatrix} = [\ A\mathbf{p}_1\ A\mathbf{p}_2\ A\mathbf{p}_3 \ldots A\mathbf{p}_n]$$

Performing the multiplication on the left hand side , we get

$$[\lambda_1\,\mathbf{p}_1\ \lambda_2\,\mathbf{p}_2\ \lambda_3\,\mathbf{p}_3 \ldots \lambda_n\,\mathbf{p}_n] = [\ A\mathbf{p}_1\ A\mathbf{p}_2\ A\mathbf{p}_3 \ldots A\mathbf{p}_n]$$

Now equating the i^{th} column of the two equal matrices above, we get

$$\lambda_i \mathbf{p}_i = A\mathbf{p}_i \tag{2.48}$$

Comparing the Equation (2.48) with (2.41), we note that λ_i is an eigen value of A and p_i is the corresponding eigen vector . So, the columns of the transformation matrix P consist of the eigen vectors of the matrix A. If the eigen values are distinct, there will be exactly n eigen vectors, which are independent of each other, so the transformation matrix P composed with the independent eigen vectors as columns will be nonsingular. Therefore, to diagonalize a matrix A, we can form a transformation matrix P whose columns are the eigen vectors of the matrix A.

Example 2.9 With reference to Example 2.8, we form P by arranging $\mathbf{x}_1$, $\mathbf{x}_2$ as two columns as shown below:

$$P = \begin{bmatrix} -1/2 & -1/3 \\ 1 & 1 \end{bmatrix}$$

and $$P^{-1} = \begin{bmatrix} -6 & -2 \\ 6 & 3 \end{bmatrix}$$

Therefore, $$B = P^{-1}AP = \begin{bmatrix} -6 & -2 \\ 6 & 3 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -6 & -5 \end{bmatrix} \begin{bmatrix} -1/2 & -1/3 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -3 \end{bmatrix}$$

2.12.2 Jordan Blocks

When the eigen vectors for a matrix are not linearly independent, it cannot be diagonalized. However, it can be transformed into Jordan canonical form. The structure of a Jordan canonical form of a matrix of dimension $k \times k$ is given below:

$$J = \begin{bmatrix} J_{q1} & 0 & 0 & . & . & 0 & 0 \\ 0 & J_{q2} & 0 & . & . & 0 & 0 \\ 0 & 0 & J_{q3} & . & . & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & J_{q(k-1)} & 0 \\ 0 & 0 & 0 & . & . & 0 & J_{qk} \end{bmatrix} \tag{2.49}$$

where $q_i \times q_i$ matrix J_{qi} are of the form,
$$J_{qi} = \begin{bmatrix} \lambda & 1 & 0 & . & . & 0 & 0 \\ 0 & \lambda & 1 & . & . & 0 & 0 \\ 0 & 0 & \lambda & . & . & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & \lambda & 1 \\ 0 & 0 & 0 & . & . & 0 & \lambda \end{bmatrix} \tag{2.50}$$

The matrices J_q are referred to as Jordan blocks of order q. It is to be noted that the eigen values λ in J_{qi} and J_{qj} need not be always different. A 6 × 6 Jordan canonical form with $q_1 = 2$, $q_2 = 3$ and $q_3 = 1$ with eigen values $\lambda_1, \lambda_1, \lambda_1, \lambda_1, \lambda_1, \lambda_6$ is shown below.

$$J = \begin{bmatrix} J_{2(\lambda_1)} & 0 & 0 & . & . & 0 & 0 \\ 0 & . & 0 & . & . & 0 & 0 \\ 0 & 0 & J_{3(\lambda_1)} & . & . & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & . & . & 0 & J_{1(\lambda_6)} \end{bmatrix} = \begin{bmatrix} \lambda_1 & 1 & 0 & . & . & 0 \\ 0 & \lambda_1 & 0 & . & . & 0 \\ 0 & 0 & \lambda_1 & 1 & . & 0 \\ . & . & . & \lambda_1 & 1 & . \\ . & . & . & . & \lambda_1 & . \\ 0 & 0 & 0 & . & . & \lambda_6 \end{bmatrix}$$

2.13 MINIMAL POLYNOMIAL FUNCTION AND COMPUTATION OF MATRIX FUNCTION USING SYLVESTER'S INTERPOLATION

Let A be $a\ n \times n$ matrix with characteristic polynomial given by :

$$f(\lambda) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \alpha_{n-2}\lambda^{n-2} + ... + \alpha_1\lambda + \alpha_0 \tag{2.51}$$

According to Cayley-Hamilton theorem, a matrix satisfies its own characteristic equation. However, the characteristic equation is not necessarily, the minimal polynomial equation. A *minimal polynomial* $\phi(\lambda)$, is a polynomial with a least degree in λ of the form :

$$\phi(\lambda) = \lambda^m + \alpha_{m-1}\lambda^{m-1} + \alpha_{m-2}\lambda^{m-2} + ... + \alpha_1\lambda + \alpha_0, \quad m \le n \tag{2.52}$$

The minimal polynomial equation that the matrix satisfies is given by:

$$\phi(A) = A^m + \alpha_{m-1}A^{m-1} + \alpha_{m-2}A^{m-2} + ... + \alpha_1 A + \alpha_0 I = 0 \tag{2.53}$$

where, I is an $n \times n$ unity matrix.

If polynomial $f_1(\lambda)$ is the greatest common factor in the adjoint matrix of $(\lambda I - A)$, then it can be shown that

$$f(\lambda) = |\ \lambda I - A\ | = f_1(\lambda)\,\phi(\lambda) \tag{2.54}$$

In general, any function of $n \times n$ matrix A can be written as

$$\phi(A) = p(A)\,\phi(A) + \alpha(A) \tag{2.55}$$

where the polynomial $\phi(A)$ is of degree $m \le n$ and $\alpha(A)$ is of degree $m - 1$ or less. So we can write $\alpha(\lambda)$ in the form :

$$\alpha(\lambda) = \alpha_0 + \alpha_1\lambda + \alpha_2\lambda^2 + \alpha_3\lambda^3 + ... + \alpha_{m-2}\lambda^{m-2} + \alpha_{m-1}\lambda^{m-1} \tag{2.56}$$

and
$$\phi(\lambda) = p(\lambda)\,\phi(\lambda) + \alpha(\lambda) \tag{2.57}$$

If we consider the case that the matrix A has m distinct eigenvalues $\lambda_1, \lambda_2, \lambda_3, ... \lambda_{m-1}, \lambda_m$, then relation (2.57) becomes:

$$\phi(\lambda) = p(\lambda)\,\phi(\lambda) + \alpha(\lambda)$$
$$= p(\lambda)(\lambda - \lambda_1)(\lambda - \lambda_2) \ldots (\lambda - \lambda_{m-1})\,(\lambda - \lambda_m) + \alpha(\lambda) \quad (2.58)$$

Now, substituting $\lambda = \lambda_i$, $i = 1 \ldots m$ in relation (2.58) and using Equation (2.56), we have the following m equations in α_i, $i = 0, 1, 2 \ldots m$

$$\alpha_0 + \alpha_1\lambda_1 + \alpha_2\lambda_1^2 + \alpha_3\lambda_1^3 + \ldots + \alpha_{m-2}\lambda_1^{m-2} + \alpha_{m-1}\lambda_1^{m-1} = \varphi(\lambda_1)$$
$$\alpha_0 + \alpha_1\lambda_2 + \alpha_2\lambda_2^2 + \alpha_3\lambda_2^3 + \ldots + \alpha_{m-2}\lambda_2^{m-2} + \alpha_{m-1}\lambda_2^{m-1} = \varphi(\lambda_2)$$
$$\alpha_0 + \alpha_1\lambda_3 + \alpha_2\lambda_3^2 + \alpha_3\lambda_3^3 + \ldots + \alpha_{m-2}\lambda_3^{m-2} + \alpha_{m-1}\lambda_3^{m-1} = \varphi(\lambda_3) \quad (2.59)$$
$$\vdots$$
$$\alpha_0 + \alpha_1\lambda_m + \alpha_2\lambda_m^2 + \alpha_3\lambda_m^3 + \ldots + \alpha_{m-2}\lambda_m^{m-2} + \alpha_{m-1}\lambda_m^{m-1} = \varphi(\lambda_m)$$

So, m unknown α_i can be solved from the above m simultaneous equation computed in terms of $\phi(\lambda_i)$, $i = 1, 2, \ldots m$. Since $\phi(A)$ is a minimal polynomial equation, $\phi(A) = 0$. So, any matrix function, $\varphi(A)$ is found as:

$$\phi(A) = p(A)\,\phi(A) + \alpha(A) = \alpha(A)$$
$$= \alpha_0 I + \alpha_1 A + \alpha_2 A^2 + \alpha_3 A^3 + \ldots + \alpha_{m-2}\, A^{m-2} + \alpha_{m-1}\, A^{m-1} \quad (2.60)$$
$$= \sum_{i=0}^{m-1} \alpha_i A^i \quad (2.61)$$

Relation of the form (2.61) can also be established when the $n \times n$ matrix A has some repeated eigen values, in which case α_i will be functions of $\phi(\lambda_i)$ as well as its derivative evaluated at the repeated eigen values.

The set of equations in (2.59) and Equation (2.60), is referred to as Sylvester's interpolation formula and is a powerful tool for computing matrix functions.

Example 2.10 As an illustration of the Sylvester's interpolation formula for computation of e^{At} , let us consider the following matrix:

$$A = \begin{bmatrix} 0 & 1 \\ -10 & -7 \end{bmatrix}$$

The characteristic polynomial, which is also the minimal polynomial here, is given by $|\lambda I - A| = \lambda^2 + 7\lambda + 10$. The eigenvalues of the matrix are found from the characteristic equation $\lambda^2 + 7\lambda + 10 = 0$ and are found as $\lambda_1 = -2$ and $\lambda_2 = -5$.

Here $m = n = 2$. Now the matrix function e^{At} may be expressed as : $e^{At} = \phi(A) = \alpha_0 I + \alpha_1 A$

The scalar coefficients α_0 and α_1 can be found from Equation (2.59) in terms of $\lambda_1 = 2$ and $\lambda_2 = 5$. Hence

$$\alpha_0 - 2\alpha_1 = e^{-2t}$$
$$\alpha_0 - 5\alpha_1 = e^{-5t}$$

The values of the coefficients are : $\alpha_1 = 1/3(e^{-2t} - e^{-5t})$ and $\alpha_0 = 1/3(5e^{-2t} - 2e^{-5t})$

Hence $e^{At} = \alpha_0 I + \alpha_1 A = \dfrac{1}{3}\,(5e^{-2t} - 2e^{-5t}) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} + \dfrac{1}{3}\,(5e^{-2t} - e^{-5t}) \begin{bmatrix} 0 & 1 \\ -10 & -7 \end{bmatrix}$

or,
$$e^{At} = \begin{bmatrix} \frac{1}{3}(5e^{-2t} - 2e^{-5t}) & \frac{1}{3}(e^{-2t} - e^{-5t}) \\ \frac{10}{3}(e^{-2t} - e^{-5t}) & \frac{1}{3}(5e^{-5t} - 2e^{-2t}) \end{bmatrix}$$

We get the same result when the state transition matrix $\phi(t) = e^{At}$ is reworked by taking the Laplace inverse transform of the resolvent matrix $\phi(s)$ in Example 3.7 in Chapter 3 [see Equation (3.87)].

MATLAB SCRIPTS


```
% Script_Sec2.3(d)
% Finding Laplace transform using MATLAB command
clear all; close all hidden
syms t s A w
F=laplace(A*sin(w*t),t,s)
```

```
% Script_Example2_1.m
% MATLAB script for finding the residues, poles and direct term
 % of a partial fraction expansion
clear all; close all hidden;syms t s
num=[1 4];                                          % numerator
den1=[1 1]; den2=[1 2]; den=conv(den1,den2);        % denominator
[R,P,K] = residue(num, den)

     % residue(num,den) finds the residues, poles and direct term of a partial
     % fraction expansion of the ratio of two polynomials num and den
     % where num and den specify the coefficients of the numerator and
     % denominator polynomials in descending powers of s. The residues are
     % returned in the column vector R, the pole locations in column vector P,
     % and the direct terms in row vector K.
```

```
% Script_Example2_3.m
% Invere Laplace transform using MATLAB script

clear all; close all hidden;
syms t s
F=(3*s+26)/(s^2+4*s+20);
a=ilaplace(F,s,t)
```

```
% Script_Example2.4.m
all clear; close all hidden;
syms t s
F=(4*s^2+19*s+24)/((s+2)^3);
a=ilaplace(F,s,t)
```

REVIEW EXERCISE

RE2.1 Find the Laplace transform of $f(t)$ given by :

$$f(t) = \begin{cases} 1, & 0 \le t \le T \\ 0, & t > T \end{cases}$$

Hints: The function $f(t)$ can be written as $f(t) = u(t) - u(t - T)$, where u is of unity strength.

Ans: The Laplace transform is given by : $F(s) = \dfrac{1}{s} - \dfrac{e^{-Ts}}{s} = \dfrac{1 - e^{-Ts}}{s}$

RE2.2 Find the poles of the function F(s) given below

$$F(s) = \frac{1}{1 - e^{-s}}$$

Ans: $s = \pm j2n\pi$ (n = 0, 1, 2, ...); **Hints:** poles are found from $e^{-s} = 1$, $s = \alpha + j\omega$

RE2.3 Find the Laplace transform of $f(t)$ defined by :

(a) $f(t) = -\dfrac{1}{4} e^{-t} + \dfrac{11}{2} te^{-3t} + \dfrac{1}{4} e^{-3t}$;

Ans: $F(s) = \dfrac{-1}{4(s+1)} + \dfrac{11}{2(s+3)^2} + \dfrac{1}{4(s+3)} = \dfrac{5s+4}{(s+1)(s+3)^2}$

(b) $f(t) = 0 \quad (t < 0)$

$= te^{-3t} \quad (t \ge 0)$

Ans: $F(s) = £[te^{-3t}] = -\dfrac{d}{ds}(s+3)^{-1} = \dfrac{1}{(s+3)^2}$

RE2.4 Find the Laplace transform of $f(t)$ given by

$f(t) = 0, \quad t < 0$

$= \sin(\omega t + \theta), \quad t \ge 0$, where θ is a constant.

Hints: $\sin(\omega t + \theta) = \sin \omega t \cos \theta + \cos \omega t \sin \theta$; **Ans:** $F(s) = \dfrac{\omega \cos\theta + s \sin\theta}{s^2 + \omega^2}$

RE2.5 Find the Laplace transform of $f(t)$ defined by

$f(t) = 0, \quad t < 0$

$= t^2 \sin \omega t,\ t \ge 0$

Hints: First find $£[\sin \omega t] = \dfrac{\omega}{s^2 + \omega^2}$, then apply the complex differentiation theorem

$$£[t^2 f(t)] = \frac{d^2}{ds^2} F(s) = \frac{-2\omega^3 + 6\omega s^2}{(s^2 + \omega^2)^3}$$

RE2.6 Compute the unit-impulse response $y(t)$ of the closed loop system given below and hence find the value of

$$\int_0^\infty y^2(t)\, dt.$$

$$\frac{Y(s)}{R(s)} = \frac{1}{s^2 + 2\delta s + 1} \ (0 < \delta < 1);$$ **Ans:** $\dfrac{1}{4\delta}$

Hints: Find Y(s) with R(s) = 1, and its inverse Laplace transform $y(t)$,

Then compute $y^2(t)$, $$y^2(t) = \frac{1}{1-\delta^2} e^{-2\delta t} \frac{1}{2}(1-\cos 2\sqrt{1-\delta^2}\, t)$$

and its Laplace transform $£\,[y^2(t)] = \frac{1}{2(1-\delta^2)}\left[\frac{1}{s+2\delta} - \frac{s+2\delta}{(s+2\delta)^2+4(1-\delta^2)}\right] = Y_1(s)$, say

Finally, $\int_0^{\infty} y^2(t)\,dt = \lim_{s\to 0}\int_0^{\infty} y^2(t)\,e^{-st}\,dt = \lim_{s\to 0} £\,[y^2(t)] = \perp \lim_{s\to 0} Y_1(s) = \frac{1}{4\delta}$, on evaluating the limit.

Chapter 2

RE2.7 Find the value of the convolution integral

$$f_1(t) * f_2(t) = \int_0^t \tau\,[1-e^{-(t-\tau)}]\,d\tau = \int_0^t (t-\tau)(1-e^{-\tau})\,d\tau$$

where

$f_1(t) = f_2(t) = 0$ for $t < 0$
$f_1(t) = t$ for $t \geq 0$
$f_2(t) = 1 - e^{-t}$ for $t \geq 0$

Hints: Find $F_1(s) = 1/s$ and $F_2(s) = 1/s(s+1)$; then find the product $F_1(s).\,F_2(s) = F(s)$ and finally $f(t)$, which is Laplace inverse transform of F(s), given by

$$f(t) = \frac{t^2}{2} - t + 1 - e^{-t}$$

RE2.8 Find the inverse Laplace transform of $F(s) = \frac{2(s+2)}{s(s^2+2s+5)}$

Hints: $$F(s) = \frac{2(s+2)}{s(s^2+2s+5)} = \frac{a_1}{s} + \frac{a_2 s + a_3}{s^2+2s+5}$$

$$= \frac{4}{5s} + \frac{-4s/5+2/5}{s^2+2s+5} = \frac{4}{5}\frac{1}{s} - \frac{4}{5}\frac{s+1}{(s+1)^2+2^2} + \frac{3}{5}\frac{2}{(s+1)^2+2^2}$$

The inverse Laplace transform of F(s) is given by

$$f(t) = \frac{4}{5} - \frac{4}{5}e^{-t}\cos 2t + \frac{3}{5}e^{-t}\sin 2t\,;\ (t \geq 0)$$

RE2.9 Obtain the inverse Laplace transform of $F(s) = \frac{2(s+2)}{s^2(s+1)(s+4)}$

Ans: $f(t) = t - \frac{3}{4} + \frac{1}{12}e^{-4t} + \frac{2}{3}e^{-t}\ (t \geq 0)$

RE2.10 Find the unit-step response of a closed loop system Y(s)/R(s) = G(s)/[1 + G(s)H(s)] with H(s) = 1 and G(s) given by

$$G(s) = \frac{5(s+20)}{s(s+5)(s+3)(s+15)}$$

Ans: The time response $y(t)$ can be found by taking the inverse Laplace transform of Y(s) :
$y(t) = 1 + 0.0009\, e^{-14.9860t} - 0.0758\, e^{-5.5424t} + 0.9000\, e^{-1.8043t} - 1.8252\, e^{-0.6673t};\ (t \geq 0)$

RE2.11 Find the Laplace transform of the following differential equation :

$$\ddot{x} + 2\dot{x} + 17x = 0 \quad x(0) = 0,\ \dot{x}(0) = 1$$

Taking the inverse Laplace transform of X(s), obtain the time solution $x(t)$.

Solution. The Laplace transform of the differential equation is

$$s^2X(s) - sx(0) - \dot{x}(0) + 2sX(s) - 2x(0) + 17X(s) = 0$$

Substituting the initial conditions and solving for X(*s*),

$$X(s) = \frac{1}{s^2 + 2s + 17}$$

The inverse Laplace transform of X(*s*) is: $x(t) = \frac{1}{4} e^{-t} \sin(4t)$

RE2.12 A mechanical system is described by : $m\ddot{x} + kx = \delta(t)$

where δ is an unit impulse and $x(0) = \dot{x}(0) = 0$

Find the resulting solution. **Ans:** $x(t) = \frac{1}{\sqrt{mk}} \sin \sqrt{\frac{k}{m}}\, t$

RE2.13 Find the transfer function Y(*s*)/ D(*s*) for the system shown in Fig. RE2.12.

Ans: $$\frac{Y(s)}{D(s)} = \frac{G_2(s)}{1 + G_1(s)\, G_2(s)H(s)}$$

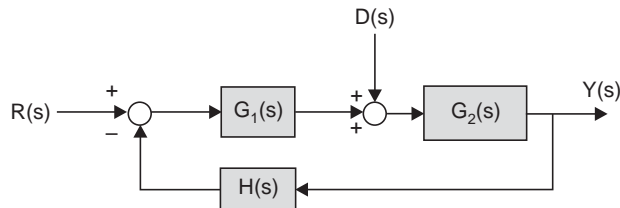



Fig. RE2.12 System with disturbance

RE2.14 The rotational velocity ω of a satellite is adjusted by changing the length, L of a beam. The transfer function between ω(*s*) and the incremental change in beam length L(*s*) is

$$\frac{\omega(s)}{L(s)} = \frac{2(s+3)}{(s+5)(s+1)^2}$$

Determine the response of the velocity ω(*t*) for a beam length change of L(*s*) = 1/(2s).

Ans: $$\omega(t) = \frac{3}{5} + \frac{1}{40} e^{-5t} - \frac{1}{2} te^{-t} - \frac{5}{8} e^{-t}$$

PROBLEMS

P2.1 Using final value theorem, find the final value of *f(t)* whose Laplace transform is given by :

$$F(s) = \frac{10}{s(s+1)}$$

Verify this result by taking the inverse Laplace transform of F(s) and letting $t \to \infty$.

P2.2 Given $F(s) = \frac{1}{(s+1)^2}$

Using the initial value theorem, find the values of $f(0_+)$ and $\dot{f}(0_+)$.

P2.3 Find the inverse Laplace transform of the following functions:

(*a*) $F(s) = \dfrac{2s+5}{s(s^2+4s+5)}$; **Ans:** $f(t) = 1 - e^{-2t}\cos t$

(*b*) $F_1(s) = \dfrac{2s+4}{s^2}$; **Ans:** $f(t) = 4t + 2$

(*c*) $F_2(s) = \dfrac{5s+1}{(s+1)(s+3)^2}$; **Ans:** $f(t) = -e^{-t} + 7te^{-3t} + e^{-3t}$

P2.4 Find the inverse Laplace transform of $F(s) = \dfrac{1}{s^2(s^2+\omega^2)}$.

P2.5 Find the solution of the following differential equation.

(*i*) $2\ddot{x} + 11\dot{x} + 5x = 0, \quad x(0) = 3, \quad \dot{x}(0) = 0$

(*ii*) $\dot{x} + 4x = \delta(t), \; x(0_) = 0$

P2.6 The following differential equations represent linear time-invariant systems, where $r(t)$ denotes the input, and $y(t)$ the output. Find the transfer function $Y(s)/R(s)$ for each of the systems.

(*a*) $\dfrac{d^3y(t)}{dt^3} + 13\dfrac{d^2y(t)}{dt^2} + 32\dfrac{dy(t)}{dt} + 20y(t) = 2\dfrac{dr(t)}{dt} + 5r(t)$

(*b*) $\dfrac{d^4y(t)}{dt^4} + 15\dfrac{d^2y(t)}{dt^2} + 2\dfrac{dy(t)}{dt} + 5y(t) = 5r(t)$

(*c*) $\dfrac{d^3y(t)}{dt^3} + 20\dfrac{d^2y(t)}{dt^2} + 10\dfrac{dy(t)}{dt} + 2y(t) + 2\int_0^t y(\tau)\,d\tau = \dfrac{dr(t)}{dt} + 5r(t)$

(*d*) $4\dfrac{d^2y(t)}{dt^2} + 2\dfrac{dy(t)}{dt} + 10y(t) = r(t) + 5r(t-1)$

P2.7 A linear time-invariant multivariable system with inputs $r_1(t)$ and $r_2(t)$ and outputs $y_1(t)$ and $y_2(t)$ is described by the following set of differential equations.

$$\frac{d^2y_1(t)}{dt^2} + 5\frac{dy_1(t)}{dt} + 2y_2(t) = r_1(t) + r_2(t)$$

$$\frac{d^2y_2(t)}{dt^2} + 4\frac{dy_1(t)}{dt} + y_1(t) - y_2(t) = r_2(t) + \frac{dr_1(t)}{dt}$$

Find the following transfer functions :

$$\left.\frac{Y_1(s)}{R_1(s)}\right|_{R_2=0}, \left.\frac{Y_2(s)}{R_1(s)}\right|_{R_2=0}, \left.\frac{Y_1(s)}{R_2(s)}\right|_{R_1=0}, \left.\frac{Y_2(s)}{R_2(s)}\right|_{R_1=0}$$

P2.8 The block diagram of an electric train control system is shown in Fig. P2.8, where the system variables are given below along with their numerical values.

$r(t)$ = Transducer voltage output corresponding to the desired train speed.
$v(t)$ = speed of train, m/sec.
M = mass of train = 25,000 kg
K = amplifier gain
K_s = gain of speed indicator = 0.1 volt/m/s

The output of the controller with a step $e_c(t) = 1, \; t \geq 0$ is measured and is described by the following expression :

$$f(t) = 50(1 - 0.2e^{-5t} - 0.5e^{-10t}), \; t \geq 0$$

(*a*) Find the transfer function $G_c(s)$ of the controller.

(*b*) Derive the open-loop transfer function V(*s*)/E(*s*) of the system with the feedback path opened.

(*c*) Derive the closed-loop transfer function V(*s*)/R(*s*) of the system.

(*d*) Assuming that K is set at a value so that the train will not run away (unstable), find the steady-state speed of the train in *ft/s* when the input is $r(t) = u_s(t)$ Volt.

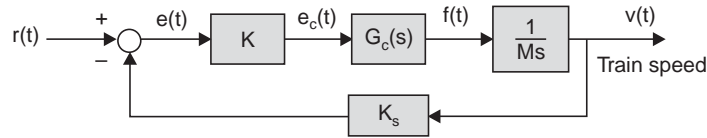



Fig. P2.8

P2.9 Draw a signal-flow graph for the following set of algebraic equations :

(*a*) $x_1 = -2x_2 - 4x_3 + 2$
$x_2 = 4x_1 - 3x_2 - x_3$
$x_3 = 5x_1 + 2x_2 - 10x_3 + 1$

(*b*) $2x_1 + 4x_2 + x_3 = -2$
$x_1 - 4x_2 + 5x_3 = 2$
$x_2 + 3x_3 = 0$

These equations should be in the form of cause-and-effect before a signal-flow graph can be drawn. Show that there are many possible signal-flow graphs for each set of equations.

P2.10 The block diagram of a feedback control system is shown in Fig. P2.10.

(*a*) Apply Mason's gain formula directly to the block diagram to find the transfer functions

$$\left.\frac{Y(s)}{R(s)}\right|_{N=0}, \left.\frac{Y(s)}{N(s)}\right|_{R=0}$$

Express Y(*s*) in terms of R(*s*) and N(*s*) when both inputs are applied simultaneously.

(*b*) Find the desired relation among the transfer functions $G_1(s)$, $G_2(s)$, $G_3(s)$, $G_4(s)$, $H_1(s)$, and $H_2(s)$ so that the output Y(*s*) is not affected by the disturbance signal N(*s*) at all.

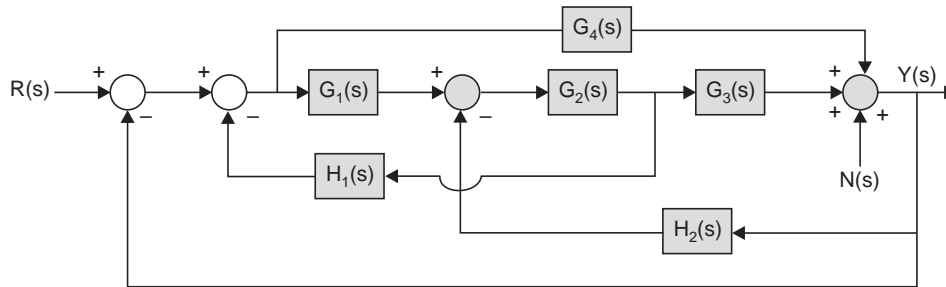



Fig. P2.10

P2.11 Linear feedback control system has the block diagram shown in Fig. P2.11.

(*a*) Find the transfer function H(*s*) so that the output Y(*s*) is not affected by the noise N(s); *i.e.*, $Y(s)/N(s)|_{R=0} = 0$.

(*b*) With H(*s*) as determined in part (*a*), find the value of K, using final-value theorem, such that the steady-state value of *e*(*t*) is equal to 0.1 when the input is a unit-ramp function, $r(t) = tu_s(t)$, $R(s) = 1/s^2$ and N(s) = 0.

$$G(s) = \frac{K(s+10)}{s(s+1)(s+5)}$$

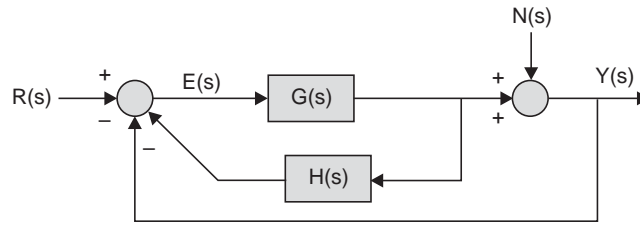



Fig. P2.11

P2.12 The block diagram of a feedback control system is shown in Fig. P2.12.

(*a*) Draw an equivalent signal-flow graph for the system.

(*b*) Find the Δ of the system by means of Mason's gain formula.

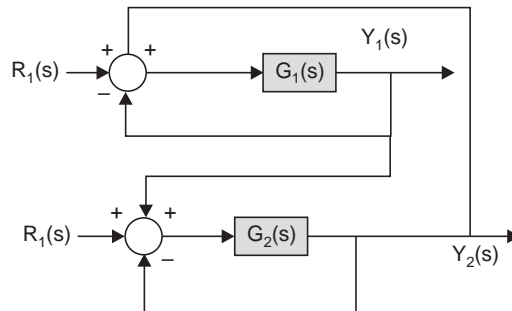



Fig. P2.12

P2.13 The block diagram of a feedback control system is shown in Fig. P2.13.

(*a*) Derive the following transfer functions :

$$\left.\frac{Y(s)}{R(s)}\right|_{N=0}, \left.\frac{Y(s)}{N(s)}\right|_{R=0}, \left.\frac{E(s)}{R(s)}\right|_{N=0}$$

(*b*) Find the transfer function $G_4(s)$ so that the output Y(*s*) is totally independent of N(*s*).

(*c*) Assuming that all the poles of *s*E(*s*) are in the left-half *s*-plane, find the steady-state value of *e*(*t*) when the input *r*(*t*) is a unit-step function with N(*t*) = 0.

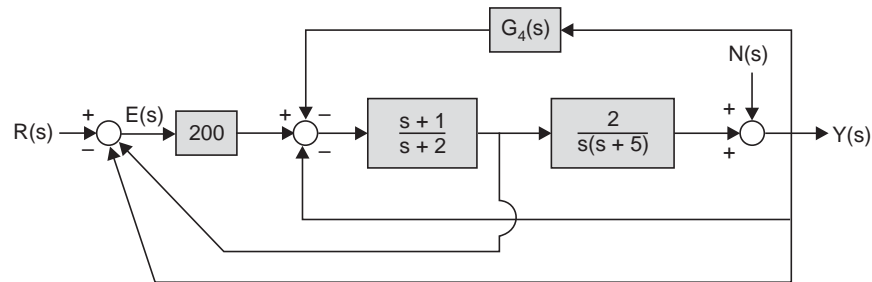



Fig. P2.13

P2.14 The position control system for a spacecraft platform is governed by the following equations:

$$\frac{d^2p}{dt^2} + 2\frac{dp}{dt} + 5p = \theta$$

$$v_1 = r - p$$

$$\frac{d\theta}{dt} = 0.6\, v_2$$

$$v_2 = 7v_1$$

The variables involved are as follows:

$r(t)$ = desired platform position
$p(t)$ = actual platform position
$v_1(t)$ = amplifier input voltage
$v_2(t)$ = amplifier output voltage
$\theta(t)$ = motor shaft position

(*a*) Draw the signal-flow diagram of the system, identifying the component parts and their transmittances.

(*b*) Find the system transfer function $P(s)/R(s)$.

CHAPTER

3

State Variable Representation and Solution of State Equations

3.1 INTRODUCTION

Modern control theory and modeling of dynamical system is almost completely dependent upon *state variable* form of representation since it is amenable to efficient solution by digital computers. In a state-variable representation an n-th order system is described in n first-order differential equations, as opposed to a single n-th order differential equation model. Before we formally present the concept of states, let us consider the solution of the first order scalar equation

$$\dot{x}(t) = ax(t) + bu(t) \tag{3.1 a}$$

which can be rearranged as

$$\dot{x}(t) - ax(t) = bu(t) \tag{3.1 b}$$

Multiplying both sides of this equation by e^{-at}, we obtain

$$e^{-at}\,[\dot{x}(t) - ax(t)] = e^{-at}\,bu(t)$$

or

$$\frac{d}{dt}\,(e^{-at}\,x(t)) = e^{-at}\,bu(t)$$

Integrating this equation between t_0 and t

$$e^{-at}\,x(t)\Big|_{t_0}^{t} = \int_{t_0}^{t} e^{-at}\,b\,u(t)\,dt = \int_{t_0}^{t} e^{-a\tau}\,b\,u(\tau)\,d\tau$$

or

$$e^{-at}\,x(t) - e^{-at_0}\,x(t_0) = \int_{t_0}^{t} e^{-a\tau}\,b\,u(\tau)\,d\tau$$

or

$$e^{-at}\,x(t) = e^{-at_0}\,x(t_0) + \int_{t_0}^{t} e^{-a\tau}\,b\,u(\tau)\,d\tau$$

or

$$x(t) = e^{a(t-t_0)}\,x(t_0) + e^{at}\int_{t_0}^{t} e^{-a\tau}\,b\,u(\tau)\,d\tau = e^{a(t-t_0)}\,x(t_0) + \int_{t_0}^{t} e^{a(t-\tau)}\,b\,u(\tau)\,d\tau \tag{3.2}$$

The first part of the solution, $e^{a(t-t_0)}\,x(t_0)$ corresponds to the homogeneous equation $\dot{x}(t) - ax(t) = 0$ while the second part, known as convolution integral, is due to the forcing function $u(t)$. The complete solution $x(t)$ can also be written, by using equivalent form of the convolution integral, as

$$x(t) = e^{a(t-t_0)}\, x(t_0) + \int_{t_0}^{t} e^{a\tau}\, bu(t-\tau)\, d\tau \tag{3.3}$$

$$= g(t-t_0)\, x(t_0) + \int_{t_0}^{t} g(\tau)\, bu(t-\tau) d\tau \tag{3.4}$$

where $g(t) = e^{at}$ is the unit impulse response of the autonomous system $\dot{x}(t) = ax(t)$. This can be readily verified from Equation (3.2) that $x(t) = e^{at} = g(t)$ with the substitution of $u(t) = 0$ and $x(t_0) = 1$ at $t_0 = 0$ (also see Sec 4.6 in Chapter 4).

From Equation (3.2) or (3.3) we see that the value of $x(t)$ can be uniquely computed at any future time t if it's initial value $x(t_0)$ is known along with current value of the forcing function $u(t)$. Similarly the solution of the second order differential equation

$$\ddot{x}(t) + a\dot{x}(t) + bx(t) = u(t) \tag{3.5}$$

may be found as the sum of the forced response, due to $u(t)$, and the solution of the homogeneous equation. For this second order system, the solution of the homogeneous equation will depend on the initial values of x and $\dot{x}$, so these two numbers, together with $u(t)$ are necessary to define completely the solution of Equation (3.5) for all time $t > 0$. In a third-order system, a knowledge of x, $\dot{x}$, and $\ddot{x}$ would be sufficient. Clearly a knowledge of x, $\dot{x}$ and $k_1 x + k_2\dot{x}$, would not be adequate, since the last expression provides no new information about the system where k_1 and k_2 are constants. In view of the above observation we can define the concept of state as follows.

Definition of state: The state of a system at any time t_0 is the minimum set of numbers $x_1(t_0), x_2(t_0),....,x_n(t_0)$, which, along with the input to the system for $t \ge t_0$, is sufficient to determine behavior of the system for all time $t \ge t_0$.

In other words, the state of a system represents the minimum amount of information that we need to know about a system at t_0 such that its future behavior can be determined without reference to the input before t_0.

The concept of system state is a physical concept and may be used to describe the behavior of a physical system. Consider a cricket ball flying through the air in a game. A player in the outfield seeing the ball's current position and estimating the velocity, mentally calculates its future location and positions himself to catch the flying ball.

It is to be noted that x and $\dot{x}$ are functions of time and hence are variables. These variables, which are capable of defining the state of the system, are designated as state variables. In general, an nth-order system is described by a collection of n state variables. The state variable representation is not unique and, in fact, there is an infinite number of choices of state variables that can correctly describe the system dynamics. However, a few of these choices are in common use because they give a mathematical advantage over others and they bear some relationship to physical variables in a system.

3.2 SYSTEM REPRESENTATION IN STATE-VARIABLE FORM

In the preceding sections, we introduced the conventional approaches to system analysis using Laplace transform and z-transforms. As an alternative, we can use the so-called *modern* approach which utilizes the *state-variable* formulation. Most practicing engineers feel more comfortable with the conventional methods since they are simpler to comprehend and are supported by a wealth of graphical techniques. The state-variable technique, on the other hand, has the following advantages for analysis and design of control systems, both continuous and discrete over the conventional methods.

1. The state-variable formulation is natural and convenient for computer solutions of continuous as well as discrete- data systems.
2. The state-variable approach allows a unified representation of digital systems with various types of sampling schemes.
3. The state-variable method allows a unified representation of single-variable and multivariable systems.
4. The state-variable method can be applied to nonlinear and time-varying systems.

In the state-variable formulation, a continuous-data system is represented by a set of first-order differential equations, called *state equations.* For a discrete-data system, when all the variables are defined in discrete time kT, the state equations are in the form of first-order *difference equations.* As mentioned earlier, a discrete-data control system often contains continuous-data as well as digital-data components, and the state equations of the system will generally consist of both first-order difference as well as first-order differential equations. For this reason, we shall begin by reviewing the state equations and their solutions of linear continuous-data systems.

Quite often the system under study is represented in block diagram form. Systems with one nonlinearity, may always be represented by an nth-order differential equation of the form

$$\frac{d^n x}{dt^n} + a_n \frac{d^{n-1}x}{dt^{n-1}} + \ldots\ldots + a_2 \frac{dx}{dt} + a_1 x = 0 \tag{3.6}$$

This may be written more conveniently as: $x^{(n)} + a_n x^{(n-1)} + \ldots + a_2\dot{x} + a_l\, x = 0$ (3.7)

Equation (3.6) or (3.7) is easily reduced to n simultaneous, first-order equations by choosing the system output (or error) and its $n - 1$ derivatives as the state variables. Thus, with x_1 equal to x, and its derivatives chosen as $n - 1$ state variables, Equation (3.6) or (3.7) becomes :

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_3(t) \\
&\vdots \\
\dot{x}_n(t) &= -\, a_n x_n(t) - a_{n-1}x_{n-1}(t) - \ldots\ldots - a_2 x_2(t) - a_1\, x_1(t)
\end{aligned} \tag{3.8}$$

This particular choice of state variables (vide Section 3.5) is referred to as the phase variables, a name that stems from the coordinates of the usual phase plane on which the behavior of a second-order system of the form of (3.8) is usually depicted.

For systems of order higher than two, the phase variables, are often a poor choice for system-state variables, because beyond the first derivative, these variables are not measurable. The choice of physical system variables overcomes these disadvantages.

The most general form of the system equations considered is

$$\dot{\mathbf{x}}(t) = \mathrm{A}\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t) \tag{3.9}$$

$$\mathbf{y}(t) = \mathrm{C}\mathbf{x}(t) \tag{3.10}$$

Here x is an n-dimensional state vector, u is an r-dimensional control vector, y is an m-dimensional output vector, A is an $n \times n$ system matrix, B is an $n \times r$ control matrix and C is an $m \times n$ output matrix. In expanded form these equations become

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \cdot \\ \cdot \\ \cdot \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & a_{22} & \cdot & \cdot & \cdot & a_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{n1} & a_{n2} & \cdot & \cdot & \cdot & a_{nn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \cdot \\ \cdot \\ \cdot \\ x_n(t) \end{bmatrix} + \begin{bmatrix} b_{11} & b_{12} & \cdot & \cdot & \cdot & b_{1r} \\ b_{21} & b_{22} & \cdot & \cdot & \cdot & b_{2r} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ b_{n1} & b_{n2} & \cdot & \cdot & \cdot & b_{nr} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \\ \cdot \\ \cdot \\ \cdot \\ u_r(t) \end{bmatrix} \quad (3.11)$$

$$\begin{bmatrix} y_1(t) \\ y_2(t) \\ \cdot \\ \cdot \\ \cdot \\ y_m(t) \end{bmatrix} = \begin{bmatrix} c_{11} & c_{12} & \cdot & \cdot & \cdot & c_{1n} \\ c_{21} & c_{22} & \cdot & \cdot & \cdot & c_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{m1} & c_{m2} & \cdot & \cdot & \cdot & c_{mn} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ \cdot \\ \cdot \\ \cdot \\ x_n(t) \end{bmatrix} \quad (3.12)$$

The state variable representation in Equations (3.11) and (3.12) allows r inputs and m outputs. Equation (3.11) is a set of n first-order differential equations and is usually referred to as the plant equation whereas Equation (3.12) represents a set of m linear algebraic equations and is referred to as the output expression. In its most general form the output expression appears as

$$\mathbf{y}(t) = \mathrm{C}\mathbf{x}(t) + \mathrm{D}\mathbf{u}(t) \quad (3.12\ a)$$

where the term $\mathrm{D}u(t)$ indicates a direct coupling of the input to the output. Since in a control system, the output power level is normally higher than that of the input, the direct coupling of the input to the output is rare and the simpler form of output expression in Equation (3.12) is adequate. However, in a communication system the presence of the term $\mathrm{D}u(t)$ is quite common.

This time-domain state-variable representation of a multiple input-multiple output system is to be contrasted with the frequency-domain transfer-function approach of classical control theory. The transfer-function representation of the above system would become simply

$$\mathrm{Y}(s) = \mathrm{G}(s)\mathrm{U}(s) \quad (3.13)$$

Here, G(s) is a transfer function matrix for multi input multi output system, since each of its elements is a transfer function between an input and output. For example, the $g_{ij}(s)$ element is the transfer function between the jth input and the ith output, that is

$$\frac{y_i(s)}{u_j(s)} = g_{ij}(s)$$

It is usual to represent graphically the transfer function by means of a block diagram, as shown in Fig. 3.1 (a) . For comparison, a block diagram of the state-variable representation of the system is shown in Fig. 3.1 (b). The diagram in Fig. 3.1(b) is nothing more than a pictorial representation of Equations (3.11) and (3.12). For the single input-single output system, the state-variable representation becomes

$$\dot{\mathbf{x}}(t) = \mathrm{A}\mathbf{x}(t) + b\mathbf{u}(t) \quad (3.14)$$

$$\mathbf{y}(t) = c'\mathbf{x}(t) \quad (3.15)$$

where the scalars u and y are the input and output respectively and b and c are n-dimensional vectors. The transfer-function representation is simply

$$\frac{\mathrm{Y}(s)}{\mathrm{U}(s)} = \mathrm{G}(s) \quad (3.16)$$

where G(s) is a scalar transfer function. A graphical representation of this system is shown in Fig. 3.2 for both the state-variable and transfer-function forms. In the classical terminology,

Equations (3.14) and (3.15) represent the fixed-plant portion of the control problem such as the mechanical load and its associated drive equipment in a position control system. Therefore G(s) is the uncompensated forward-path transfer function, while the control input $u(t)$ is the classical actuating signal, such as the armature current in a DC positioning system. The broad arrow on block diagrams, as in Figs. 3.1 and 3.2 has been used to indicate vector quantities. The most obvious difference between the state-variable and transfer-function approaches is the suppression of the state vector in the transfer function approach. This is quite expected since the transfer function approach is concerned only with the input-output characteristics of a system while the state-variable approach provides more detailed and complete description of the internal behavior of the system. It will be shown shortly that the transfer-function representation of a system is equivalent to the state variable representation only under certain circumstances. When the two representations are equivalent, one form can be obtained from the other. Since the transfer-function method specifies only the input-output relations of the system, it is unique for a system for the given pair of input-output. But in general, there is, an infinite number of state-variable representations for a given transfer function.

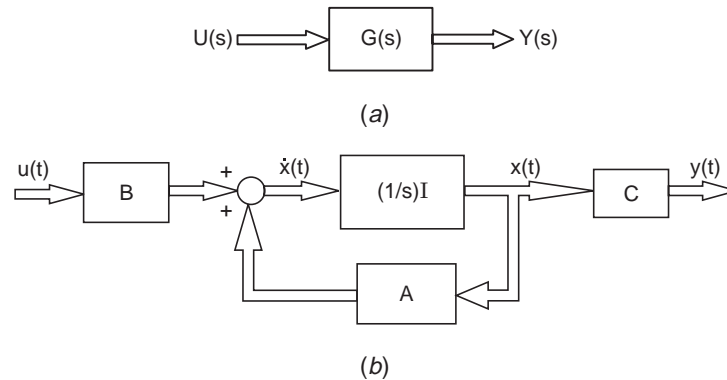



Fig. 3.1 Block diagram of a multi input-multi output system
(*a*) transfer function representation; (*b*) state-variable representation

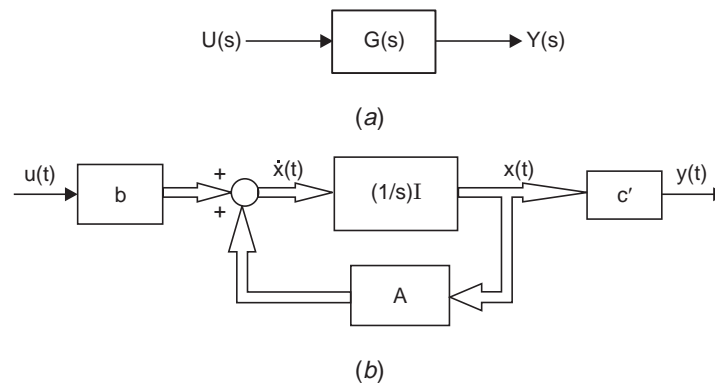



Fig 3.2 Block diagram of a single input-single output system
(a) transfer function representation; (b) state-variable representation

3.3 CONCEPTS OF CONTROLLABILITY AND OBSERVABILITY

Two new concepts termed as *controllability* and *observability* are to be defined in connection with a control system, in order to establish the equivalence of state variable and transfer

function representation. This concept can be demonstrated only in terms of the state-variable representation of a system and cannot be used for transfer-function representation since, as will be discovered, the transfer function representation is valid only if it satisfies *controllability* and *observability* conditions.

The concept of controllability and observability were first introduced by Kalman [9] and are necessary to establish the validity of the classical transfer function methods.

Definition of controllability: A system is said to be controllable if any initial state $x(t_0)$ can be transferred to any final state $x(t_f)$ in a finite time, $t_f \geq 0$, by some control u.

Controllability implies, as the term suggests, the ability of the control input to affect each state variable. Observability on the other hand is concerned with the ability of each state variable to influence the output of the system.

Definition of observability: A system is said to be observable if every state $x(t_0)$ can be exactly determined from measurements of the output y over a finite interval of time $t_0 \leq t \leq t_f$.

The following test can be used to establish the controllability and observability of a system in any state-variable representation.

Theorem 3.1 The nth-order multiple input-multiple output system

$$\dot{\mathbf{x}}(t) = \mathrm{A}\,\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t) \tag{3.17}$$

$$\mathbf{y}(t) = \mathrm{C}\mathbf{x}(t) \tag{3.18}$$

is

(*a*) *Controllable* if and only if the $n \times nr$ composite matrix

$$[\mathrm{B}\ \mathrm{AB}\ \mathrm{A}^2\mathrm{B}\ \ldots.\ \mathrm{A}^{n-1}\mathrm{B}\ ] \tag{3.19}$$

is of rank n;

(*b*) *Observable* if and only if the $n \times$ nm composite matrix

$$[\mathrm{C}'\ \mathrm{A}'\mathrm{C}'\ \mathrm{A}'^2\mathrm{C}'\ \mathrm{A}'^3\mathrm{C}'\ \ldots.\ \mathrm{A}'^{\,n-1}\mathrm{C}'] \tag{3.20}$$

is of rank n.

In the case of the single input-single output system, this test becomes:

Theorem 3.2 The nth-order single input-single output system

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) + b\mathbf{u}(t) \tag{3.21}$$

$$\mathbf{y}(t) = c'\mathbf{x}(t) \tag{3.22}$$

is

(*a*) *Controllable* if and only if the $n \times n$ composite matrix

$$[b\ \mathrm{A}b\ \mathrm{A}^2 b \ldots\ldots\ \mathrm{A}^{n-1}b] \tag{3.23}$$

is nonsingular; *i.e.*, its determinant is nonzero.

(*b*) *Observable* if and only if the $n \times n$ composite matrix

$$[c\ \mathrm{A}'c\ \mathrm{A}'^2 c\ \mathrm{A}'^3 c\ \ldots.\ \mathrm{A}'^{\,n-1}c] \tag{3.24}$$

is nonsingular.

Proof of Theorem 3.1

We shall prove only **Theorem 3.1**. In order to prove it, we note that the solution of Equation (3.17) can be written, by analogy to the solution of the scalar equation as given by (3.2), as

$$\mathbf{x}(t) = e^{\mathrm{A}(t-t_0)}\,\mathbf{x}(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}\,\mathrm{B}u(\tau)\,d\tau \tag{3.25}$$

where A is an $n \times n$ system matrix and B is an $n \times r$ input matrix.

Choose the initial time $t_0 = 0$, the final time as t_f and the final state, without any loss of generality, as

$$\mathbf{x}(t_f) = 0$$

then from Equation (3.25) we have

$$\mathbf{x}(0) = -\int_0^{t_f} e^{-A\tau} B\, u(\tau)\, d\tau \tag{3.26}$$

Using the Sylvester's interpolation formula (see Section 2.13 in Chapter 2) for the matrix function $e^{A\tau}$ as

$$e^{-A\tau} = \sum_{k=0}^{n-1} \alpha_k(\tau) A^k \tag{3.27}$$

Equation (3.26) can be written as

$$\mathbf{x}(0) = -\sum_{k=0}^{n-1} A^k B \int_0^{t_f} \alpha_k(\tau)\, u(\tau)\, d\tau \tag{3.28}$$

For each k, the integral will be a constant given by

$$-\int_0^{t_f} \alpha_k(\tau)\, u(\tau)\, d\tau = \beta_k$$

Therefore, the Equation (3.28) reduces to

$$\mathbf{x}(0) = \sum_{k=0}^{n-1} A^k B \beta_k \tag{3.29}$$

or

$$\mathbf{x}(0) = [B\ AB\ A^2B \ldots A^{n-1} B] \begin{bmatrix} \beta_0 \\ \beta_1 \\ \cdot \\ \cdot \\ \beta_{n-1} \end{bmatrix} \tag{3.30}$$

Therefore the sequence of control signal $\beta_0, \beta_1 \ldots\ldots \beta_{n-1}$ can be uniquely computed from Equation (3.30) that will transfer the initial state $x(0)$ to final state $x(t_f)$ if the rank of the $n \times nr$ composite matrix

$$[B\ AB\ A^2B \ldots\ A^{n-1}B] \text{ is } n.$$

Proof of the second part of Theorem 3.1

Set the control input $u(t)$ to zero in Equation (3.17) since it is known for this case. So with u = 0, the Equations (3.17) and (3.18) become

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t) \tag{3.31a}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) \tag{3.31b}$$

Using the solution for $x(t)$ from Equation (3.25) and with $t_0 = 0$, $y(t)$ is written as

$$y(t) = Cx(t) = Ce^{At}\, x(0) \tag{3.32}$$

Chapter 3

Now using power series expansion for e^{At} from Equation (3.27), $y(t)$ is written as

$$y(t) = \sum_{k=0}^{n-1} \alpha_k(t)\, CA^k x(0) \tag{3.33}$$

or, $y(t) = \alpha_0(t)\, Cx(0) + \alpha_1(t)\, CAx(0) + \alpha_2(t)CA^2 x(0) + \alpha_3(t)CA^3 x(0) + \ldots \alpha_{n-1}(t)CA^{n-1}x(0)$ (3.34)

or
$$y(t) = [\alpha_0 \quad \alpha_1 \quad \alpha_2 \ldots \alpha_{n-1}] \begin{bmatrix} C \\ CA \\ CA^2 \\ \ldots \\ CA^{n-1} \end{bmatrix} x(0) \tag{3.35}$$

Therefore, measuring the output vector over the interval $0 \le t \le t_f$, $x(0)$ is uniquely determined if the rank of the $nm \times n$ matrix shown below is n

$$\begin{bmatrix} C \\ CA \\ CA^2 \\ \ldots \\ CA^{n-1} \end{bmatrix}$$

Since the rank of a matrix remains invariant as that of its transpose, the above matrix may also be put in the form as :

$$[C' \; A'C' \; A'^2C' \; A'^3C' \; A'^4C' \; \ldots \; A'^{n-1}C'] \tag{3.36}$$

This completes the proof of Theorem 3.1.

Example 3.1 Let us consider the following state-variable representation of a system.

$$\dot{\mathbf{x}}(t) = \begin{bmatrix} -5 & 0 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & -4 \end{bmatrix} \mathbf{x}(t) + \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \mathbf{u}(t); \quad \mathbf{y}(t) = [2 \quad 0 \quad 4] \times (t)$$

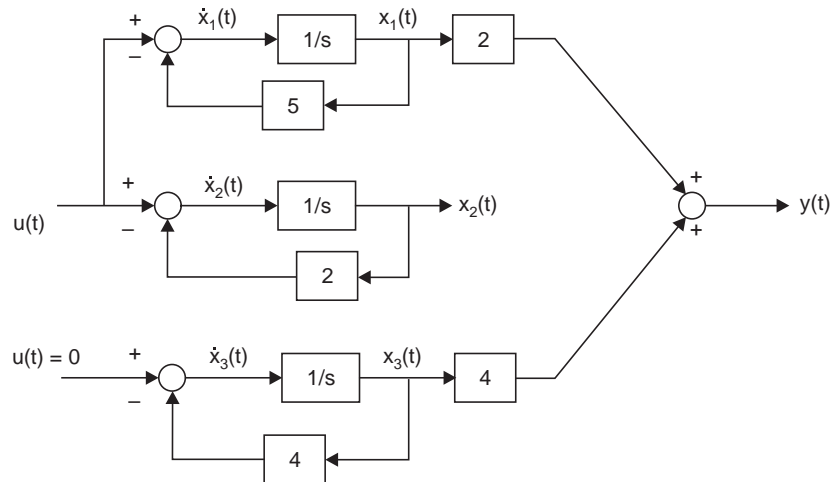



Fig. 3.3 Block diagram for the system of Example 3.1

A simulation diagram representation for this third order system is shown in Fig. 3.3, where 1/*s* represents integration and the output of the integrators are states. We observe from the Fig. 3.3 that the state $x_1(t)$ can be controlled by the input $u(t)$ and information about $x_1(t)$ is also available from the measurement of $y(t)$. The input-output transfer function for this system is obviously

$$\frac{Y(s)}{U(s)} = G(s) = \frac{2}{s+5}$$

Hence we see that when the actual system is third-order, the transfer function represents only the first-order controllable and observable part of the system.

For the system in Example 3.1, where the system matrix is in a diagonal form, the controllable and observable states can be readily identified by inspection of the simulation diagram. But for a general system representation, we have to apply the Theorems 3.1 and 3.2 for examining the controllability and observability of the system.

Example 3.2 As illustration of the application of these testing let us consider again the system of Example 3.1. In this case the system is single input, single output and hence Theorem 3.2 is applicable.

For this system $Ab = \begin{bmatrix} -10 \\ 0 \\ -4 \end{bmatrix}$ and $A^2b \begin{bmatrix} 50 \\ 0 \\ 16 \end{bmatrix}$ and the matrix (3.23) becomes

$$[b \quad Ab \quad A^2b] = \begin{bmatrix} 1 & -10 & 50 \\ 0 & 0 & 0 \\ 1 & -4 & 16 \end{bmatrix}$$

It is obvious that the matrix is singular, since it has, an all-zero row, and therefore the system is not controllable.

To check observability we form the matrix (3.24)

$$[c \quad A'c \quad A'^2c] = \begin{bmatrix} 1 & -5 & 25 \\ 4 & -8 & 16 \\ 0 & 0 & 0 \end{bmatrix}$$

which is also singular. We therefore conclude that the system is neither controllable nor observable, as we had previously seen from the simulation diagram.

It is to be observed that the rank of matrix (3.19) or (3.23) is equal to the number of controllable states, while the rank of matrix (3.20) or (3.24) is equal to the number of observable states.

All systems treated here are assumed to be both controllable and observable. This assumption is made, first of all, because almost all practical systems are controllable and observable.

The reader may wonder why the transfer-function approach is used at all since the state-variable representation provides a more complete description of the system. The simple truth is that some of the practical results of modern control theory can be most compactly and usefully presented in terms of transfer functions.

3.4 TRANSFER FUNCTION FROM STATE-VARIABLE REPRESENTATION

Having established the conditions for the equivalence of the state-variable representation with that of the transfer-function, we are interested to find one representation from the other by

finding their relationship. Let us consider first the problem of determining the transfer function of a system given the state variable representation

$$\dot{\mathbf{x}}(t) = \mathrm{A}\,\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t) \tag{3.37}$$

$$\mathbf{y}(t) = \mathrm{C}\mathbf{x}(t) \tag{3.38}$$

since the transfer-function representation is expressed in the frequency domain, we begin by taking the Laplace transform of both equations, assuming as usual in transfer-function determination that the initial conditions on x are all zero.

$$s\mathrm{X}(s) = \mathrm{AX}(s) + \mathrm{BU}(s) \tag{3.39}$$

$$\mathrm{Y}(s) = \mathrm{CX}(s) \tag{3.40}$$

Grouping the two X(s) terms in Equation (3.39) we have

$$(s\mathrm{I} - \mathrm{A})\mathrm{X}(s) = \mathrm{BU}(s) \tag{3.41}$$

where the identity matrix has been introduced to allow the indicated multiplication compatible. Now, pre-multiplying both sides of the above equation by $(s\mathrm{I} - \mathrm{A})^{-1}$, we get

$$\mathrm{X}(s) = (s\mathrm{I} - \mathrm{A})^{-1}\,\mathrm{BU}(s)$$

We substitute this result in Equation (3.40) to obtain

$$\mathrm{Y}(s) = \mathrm{C}(s\mathrm{I} - \mathrm{A})^{-1}\,\mathrm{BU}(s)$$

Comparing this relation between Y(s) and U(s) with the Equation (3.13) we find that the transfer function matrix G(s) as:

$$\mathrm{G}(s) = \mathrm{C}(s\mathrm{I} - \mathrm{A})^{-1}\,\mathrm{B} \tag{3.42}$$

For the single input-single output case, this result reduces to

$$\mathrm{G}(s) = c'(s\mathrm{I} - \mathrm{A})^{-1}\,b \tag{3.43}$$

The matrix $(s\mathrm{I} - \mathrm{A})^{-1}$ is commonly referred to as the resolvent matrix and is designated by $\phi(s)$,

$$\phi(s) = (s\mathrm{I} - \mathrm{A})^{-1} \tag{3.44}$$

In terms of this notation Equations 3.42 and 3.43 become

$$\mathrm{G}(s) = \mathrm{C}\,\phi(s)\,\mathrm{B} \tag{3.45}$$

and

$$\mathrm{G}(s) = c'\,\phi(s)\,b \tag{3.46}$$

Example 3.3 Consider the system represented by the equations

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ -10 & -7 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t); \quad y(t) = [1\ 0]\ x(t)$$

The matrix $s\mathrm{I} - \mathrm{A}$ in this example becomes

$$(s\mathrm{I} - \mathrm{A}) = \begin{bmatrix} s & -1 \\ 10 & s+7 \end{bmatrix}$$

and its inverse is found as

$$\phi(s) = (s\mathrm{I} - \mathrm{A})^{-1} = \frac{\mathrm{adj}\,(s\mathrm{I} - \mathrm{A})}{\det\,(s\mathrm{I} - \mathrm{A})} = \frac{\begin{bmatrix} s+7 & 1 \\ -10 & s \end{bmatrix}}{s^2 + 7s + 10} \tag{3.47}$$

The transfer function, therefore, is given by :

$$\mathrm{G}(s) = c'\,\phi(s)\,b = \frac{[1\ \ 0]\begin{bmatrix} s+7 & 1 \\ -10 & s \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix}}{s^2 + 7s + 10} = \frac{1}{s^2 + 7s + 10}$$

In the above example, we observe that the determinant of the matrix (sI – A) is equal to the denominator polynomial of G(s). This is always true for single input - single output systems. Although Equations (3.45) and (3.46) provide a direct method for determining the transfer function of a system from a state-variable representation of the system, it is generally not the most efficient method. This is due to the fact that one must invert the matrix (sI – A). Because of this problem, it is often easier to obtain the transfer function by carrying out block diagram reductions or equivalently signal flow-graph techniques on the block diagram of the state-variable representation.

3.4.1 Computation of Resolvent Matrix from Signal Flow Graph

Consider the system

$$\dot{\mathbf{x}}(t) = \mathrm{A}\,\mathbf{x}(t) \tag{3.48}$$

Taking Laplace Transform of Equation (3.48) we have

$$s\mathrm{X}(s) - x(0) = \mathrm{AX}(s) \tag{3.49}$$

or
$$(s\mathrm{I} - \mathrm{A})\,\mathrm{X}(s) = x(0)$$

or
$$\mathrm{X}(s) = (s\mathrm{I} - \mathrm{A})^{-1}\,x(0) = \phi(s)\,x(0) \tag{3.50}$$

Equation (3.50) is now written in expanded form for a second order system as follows:

$$\begin{bmatrix} \mathrm{X}_1(s) \\ \mathrm{X}_2(s) \end{bmatrix} = \begin{bmatrix} \phi_{11}(s) & \phi_{12}(s) \\ \phi_{21}(s) & \phi_{22}(s) \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} \tag{3.51}$$

With $x_2(0) = 0$, we get $\mathrm{X}_1(s) = \phi_{11}(s)\,x_1(0)$

and $\mathrm{X}_2(s) = \phi_{21}(s)\,x_1(0)$

Hence,
$$\phi_{11}(s) = \left.\frac{\mathrm{X}_1(s)}{x_1(0)}\right|_{x_2(0)=0} \tag{3.52}$$

and
$$\phi_{21}(s) = \left.\frac{\mathrm{X}_2(s)}{x_1(0)}\right|_{x_2(0)=0} \tag{3.53}$$

Similarly,
$$\phi_{12}(s) = \left.\frac{\mathrm{X}_1(s)}{x_2(0)}\right|_{x_1(0)=0} \tag{3.54}$$

and
$$\phi_{22}(s) = \left.\frac{\mathrm{X}_2(s)}{x_2(0)}\right|_{x_1(0)=0} \tag{3.55}$$

For a given second order system matrix A, one can find the elements of the resolvent matrix using the relations (3.52) – (3.55), which does not require the computation of the inverse of the matrix (sI – A). The concept may be illustrated by the following example.

Example 3.4 With reference to the system in (3.48),

let $\mathrm{A} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix}$, so that (3.49) becomes

$$\begin{bmatrix} s\ \mathrm{X}_1(s) \\ s\ \mathrm{X}_2(s) \end{bmatrix} - \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -a_2 & -a_1 \end{bmatrix} \begin{bmatrix} \mathrm{X}_1(s) \\ \mathrm{X}_2(s) \end{bmatrix}$$

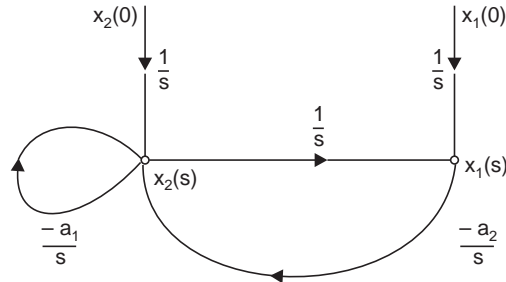



Fig. 3.4 Signal flow graph for Equations (3.56) and (3.57)

After rearrangement and division by $s \neq 0$ we get

$$X_1(s) = \frac{x_1(0)}{s} + \frac{X_2(s)}{s} \tag{3.56}$$

$$X_2(s) = \frac{x_2(0)}{s} - \frac{a_2 X_1(s)}{s} - \frac{a_1 X_2(s)}{s} \tag{3.57}$$

The signal flow diagram for the above pair of equations is shown in Fig. 3.4.

Now we can compute the transfer function between $X_1(s)$ and $x_1(0)$ with $x_2(0) = 0$ by Mason's Gain formula (vide Equation (2.29) in Chapter 2). Using relation (3.52) we have

$$\phi_{11}(s) = \left.\frac{X_1(s)}{x_1(0)}\right|_{x_2(0)=0} = \frac{M_1 \Delta_1}{\Delta}$$

where $$\Delta = 1 - \left(-\frac{a_2}{s^2} - \frac{a_1}{s}\right) = 1 + \frac{a_1}{s} + \frac{a_2}{s^2}$$

$$M_1 = \frac{1}{s} \; ; \Delta_1 = 1 - \left(-\frac{a_1}{s}\right) = 1 + \frac{a_1}{s}$$

$\therefore$ $$\phi_{11}(s) = \frac{\frac{1}{s}\left(1 + \frac{a_1}{s}\right)}{1 + \frac{a_1}{s} + \frac{a_2}{s^2}} = \frac{s + a_1}{s^2 + a_1 s + a_2}$$

Similarly $$\phi_{21}(s) = \left.\frac{X_2(s)}{x_1(0)}\right|_{x_2(0)=0} = \frac{M_1 \Delta_1}{\Delta}$$

where $$M_1 = -\frac{a_2}{s^2}, \Delta = 1 + \frac{a_1}{s} + \frac{a_2}{s^2}, \; \Delta_1 = 1$$

$\therefore$ $$\phi_{21}(s) = \frac{-\frac{a_2}{s^2}}{1 + \frac{a_1}{s} + \frac{a_2}{s^2}} = \frac{-a_2}{s^2 + a_1 s + a_2}$$

$$\phi_{12}(s) = \left.\frac{X_1(s)}{x_2(0)}\right|_{x_1(0)=0} = \frac{\frac{1}{s^2} \cdot 1}{\Delta} = \frac{1}{s^2 + a_1 s + a_2}$$

and $$\phi_{22}(s) = \left.\frac{X_2(s)}{x_2(0)}\right|_{x_1(0)=0} = \frac{\frac{1}{s}\,.\,1}{\Delta} = \frac{s}{s^2 + a_1 s + a_2}$$

So the resolvent matrix is given by $\phi(s) = \dfrac{1}{s^2 + a_1 s + a_2}\begin{bmatrix} s + a_1 & 1 \\ -a_2 & s \end{bmatrix}$

Note that the above result is identical with that in relation (3.47) with $a_1 = 7$, $a_2 = 10$

3.5 STATE VARIABLE REPRESENTATION FROM TRANSFER FUNCTION

In Section 3.4 we have shown how to get the transfer function model of a linear continuous system when its state-variable form is available and is completely controllable and observable. We shall now take up the issue of getting the state-variable model when the transfer function model is available. Since the state-variable representation is not unique, there are, theoretically, an infinite number of ways of writing the state equations. We shall present here one method for deriving a set of continuous state variable representation from the transfer function. Analogous procedure may be followed for writing the discrete state equation from pulse transfer function in z domain.

The transfer function of single-input-single-output system of the form:

$$G(s) = \frac{a_{n-1}\, s^{n-1} + a_{n-2}\, s^{n-2} + \ldots\ldots + a_1 s + a_0}{s^n + b_{n-1}\, s^{n-1} + \ldots\ldots + b_1 s + b_0} \tag{3.58}$$

can be written, after introducing an auxiliary variable E (s), as:

$$\frac{Y(s)}{U(s)} = G(s) = \frac{a_{n-1}\, s^{n-1} + a_{n-2}\, s^{n-2} + \ldots\ldots + a_1 s + a_0}{s^n + b_{n-1}\, s^{n-1} + \ldots\ldots + b_1 s + b_0}\,\frac{E(s)}{E(s)} \tag{3.59}$$

We now let

$$Y(s) = (a_{n-1}\, s^{n-1} + \ldots + a_1 s + a_0)\, E(s) \tag{3.60}$$

$$U(s) = (s^n + b_{n-1} s^{n-1} + \ldots + b_1 s + b_0)\, E(s) \tag{3.61}$$

From Theorem 2.5 of Laplace transform in Chapter 2, we note the following relations between the variables in the s domain and time domain with zero initial conditions

$$E(s) \longrightarrow e(t)$$

$$sE(s) \longrightarrow \dot{e}(t)$$

$$s^2E(s) \longrightarrow \ddot{e}(t)$$

Under this correspondence we define the state variables

$$\begin{aligned} x_1(t) &= e(t) \\ x_2(t) &= \dot{x}_1(t) = \dot{e}(t) \\ x_3(t) &= \dot{x}_2(t) = \ddot{e}(t) \\ &\vdots \\ x_n(t) &= \dot{x}_{n-1}(t) = e^{(n-1)}(t) \end{aligned} \tag{3.62}$$

From Equations (3.61) and (3.62) we obtain the state equations

$$\begin{aligned}
\dot{x}_1(t) &= x_2(t) \\
\dot{x}_2(t) &= x_3(t) \\
\dot{x}_3(t) &= x_4(t) \\
&\vdots \\
\dot{x}_n(t) &= -b_0 x_1(t) - b_1 x_2(t) - b_2 x_3(t) \ldots\ldots - b_{n-1} x_n(t) + u(t)
\end{aligned} \tag{3.63}$$

In matrix notation this becomes

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ . \\ . \\ . \\ . \\ \dot{x}_n(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 1 & 0 & 0 & . & . & . & 0 \\ 0 & 0 & 0 & 1 & 0 & . & . & . & 0 \\ . \\ . \\ . \\ -b_0 & -b_1 & -b_2 & -b_3 & -b_4 & . & . & . & -b_{n-1} \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ . \\ . \\ x_n(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ . \\ . \\ 1 \end{bmatrix} u(t) \tag{3.64}$$

In compact form, it is written as: $\dot{\mathbf{x}}(t) = \mathrm{A}\,\mathbf{x}(t) + \mathrm{B}\mathbf{u}(t)$ (3.65)

The output equation is obtained from (3.60) as

$$\mathbf{y}(t) = [a_0\ a_1 \ldots a_{n-1}] \begin{bmatrix} x_1(t) \\ x_2(t) \\ . \\ . \\ . \\ x_n(t) \end{bmatrix} \tag{3.66}$$

which may be written compactly as $\mathbf{y}(t) = \mathrm{C}\mathbf{x}(t)$ (3.67)

Hence Equations (3.65) and (3.67) are a set of state equations for the continuous system described by transfer function in Equation (3.58).

Another convenient and useful representation of the continuous system is the signal flow graph or the equivalent simulation diagram. These two forms can be derived, after dividing both the numerator and denominator of Equation (3.59) by s^n :

$$\mathrm{G}(s) = \frac{\mathrm{Y}(s)}{\mathrm{U}(s)} = \frac{a_{n-1}\, s^{-1} + a_{n-2}\, s^{-2} + \ldots\ldots + a_1 s^{1-n} + a_0 s^{-n}}{1 + b_{n-1}\, s^{-1} + \ldots\ldots + b_1 s^{1-n} + b_0 s^{-n}} \, \frac{\mathrm{E}(s)}{\mathrm{E}(s)} \tag{3.68}$$

From this expression we obtain two equations

$$\mathrm{Y}(s) = (a_{n-1} s^{-1} + a_{n-2} s^{-2} + \ldots\ldots\ldots\ldots\ldots\ldots + a_1 s^{-n+1} + a_0 s^{-n})\, \mathrm{E}(s) \tag{3.69}$$

and $$\mathrm{U}(s) = (1 + b_{n-1} s^{-1} + \ldots\ldots\ldots\ldots\ldots\ldots\ldots + b_1 s^{1-n} + b_0 s^{-n})\, \mathrm{E}(s) \tag{3.70}$$

Equation (3.70) can be rewritten in the form

$$\mathrm{E}(s) = \mathrm{U}(s) - b_{n-1} s^{-1} \mathrm{E}(s) - \ldots\ldots\ldots - b_1 s^{1-n}\, \mathrm{E}(s) - b_0 s^{-n} \mathrm{E}(s) \tag{3.71}$$

Equations (3.69) and (3.71) may be used to draw the signal flow graph shown in Fig. 3.5, whose transfer function is given by Equation (3.58).

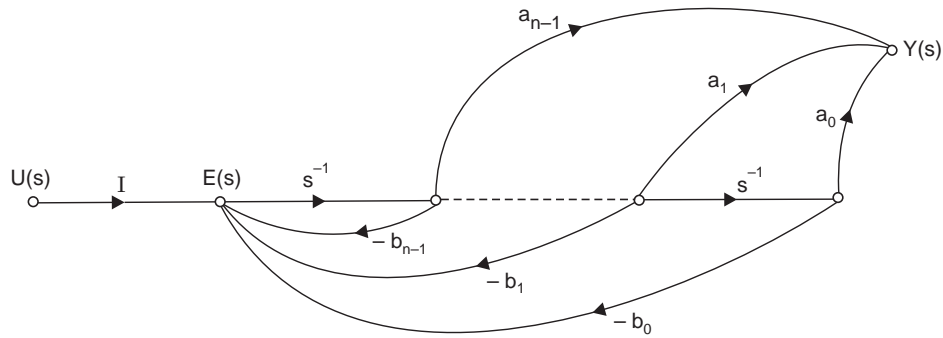



Fig. 3.5 Signal flow graph representation of Equation (3.58)

In signal flow graph the term $s^{-1} = 1/s$ represents pure integration.

The signal flow graph of Fig. 3.5 can also be represented by the equivalent simulation diagram, with the states indicated as in Fig. 3.6.

Noting the structure of the signal flow graph in Fig. 3.5 and its association with the numerator and denominator polynomials represented by Equations (3.64) and (3.66) respectively, it is apparent that the signal flow graph or simulation diagram can be obtained by inspection of the transfer function in Equation (3.58). If, on the other hand, the signal flow graph is drawn from the state equation, one may use Manson's gain formula to find transfer function instead of using relation (3.46) that necessitates the computation of inverse of a matrix. The structure of Fig. 3.5, together with Equations (3.64) and (3.66), is referred to as the phase variable canonical form of system representation.

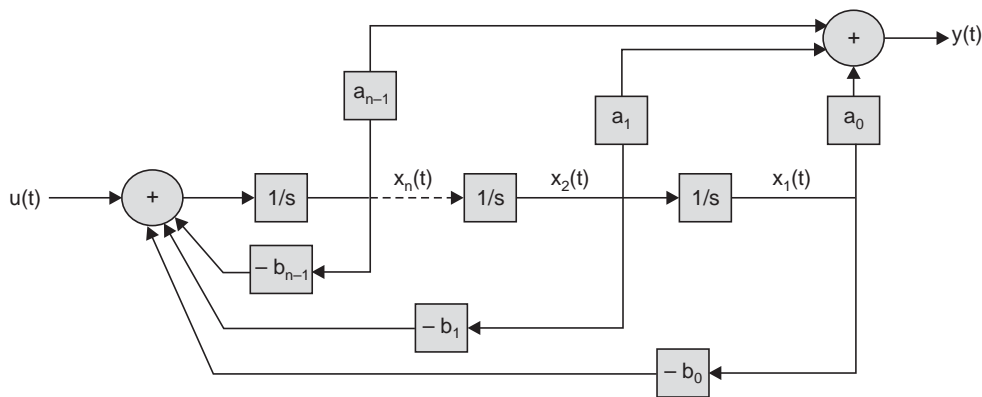



Fig. 3.6 Simulation diagram equivalent to the signal flow graph in Fig. 3.5

Another standard form called observer canonical form is shown in Fig. 3.7. The equivalence of the system in Fig. 3.7 to the Equation (3.58) may be established by computing the transfer function Y(s)/U(s) from the figure.

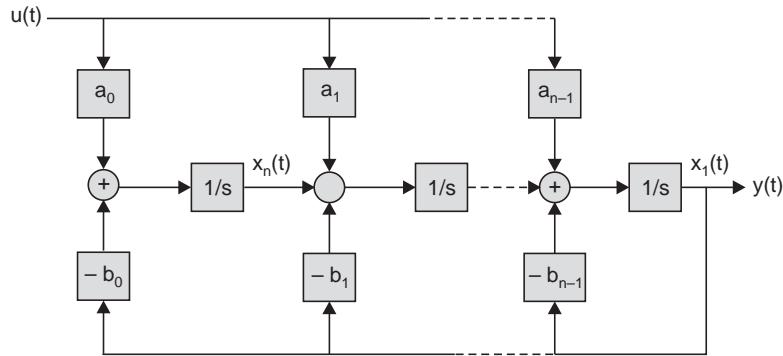



Fig. 3.7 Observer canonical form

Example 3.5 Given the following transfer function, we want to derive the signal flow graph and the corresponding state equations:

$$\mathrm{G}(s) = \frac{\mathrm{Y}(s)}{\mathrm{U}(s)} = \frac{s^2 + 5s + 4}{s^3 + 14s^2 + 48s + 80} \tag{3.72}$$

After multiplying numerator and denominator with E(s) as before and comparing this expression with (3.58) and Fig. 3.5, the signal flow graph can be derived by inspection, as shown in Fig. 3.8. The outputs of the integrators are chosen as the states. The state equations can now be derived from Fig. 3.8 or from a comparison of the given transfer function with Equations (3.58), (3.64), and (3.66). The resulting state equations are :

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \dot{x}_3(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -80 & -48 & -14 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} u(t) \tag{3.73}$$

$$y(t) = [4\ 5\ 1] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix}$$

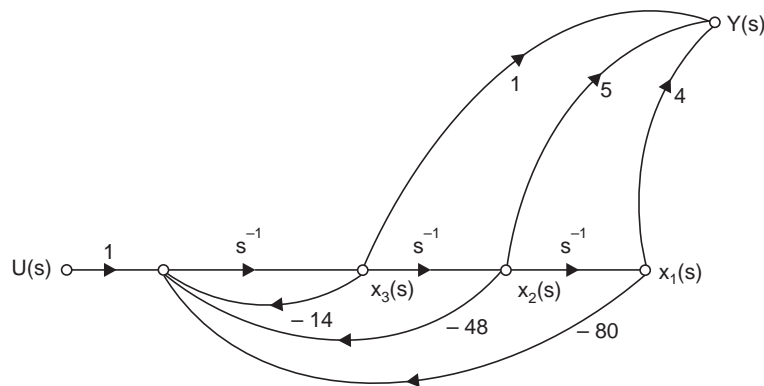



Fig. 3.8 Signal flow graph for the system in Example 3.5

When the numerator and denominator are of the same order, the output in the state variable representation needs a little bit of simplification and should be arranged as illustrated in the following example.

Example 3.6 Let us consider the following transfer function whose numerator and denominator are both third orders.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{s^3 + b_2 s^2 + b_1 s + b_0} \tag{3.74}$$

The signal flow diagram is shown in Fig. 3.9. From the flow graph it is found that

$$Y(s) = a_0\, x_1(s) + a_1 x_2(s) + a_2\, x_3(s) + a_3\, E(s)$$

But $\qquad E(s) = U(s) - b_2\, x_3\,(s) - b_1 x_2(s) - b_0\, x_1\,(s)$

From these two equations we obtain

$$Y(s) = a_3\, U(s) + (a_2 - a_3\, b_2)\, x_3\,(s) + (a_1 - a_3\, b_1)\, x_2(s) + (a_0 - a_3 b_0)\, x_1(s)$$

Hence the output state equation in time domain is given by

$$y(t) = [(a_0 - a_3 b_0) \quad (a_1 - a_3 b_1) \quad (a_2 - a_3 b_2)] \begin{bmatrix} x_1(t) \\ x_2(t) \\ x_3(t) \end{bmatrix} + a_3\, u(t) \tag{3.75}$$

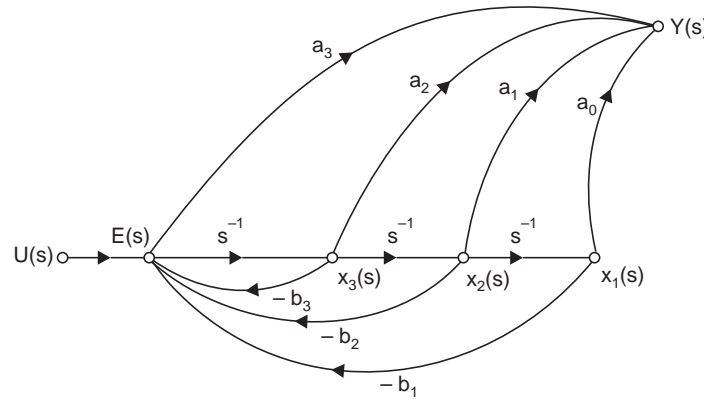



Fig. 3.9 Signal flow graph for the system in Example 3.6

3.6 SOLUTION OF STATE EQUATION AND STATE TRANSITION MATRIX

We reproduce below relations (3.9) and (3.12*a*), representing the most general form of state equations :

$$\dot{\mathbf{x}}(t) = A\, \mathbf{x}(t) + B\mathbf{u}(t) \tag{3.76}$$

$$\mathbf{y}(t) = C\mathbf{x}(t) + D\mathbf{u}(t) \tag{3.77}$$

The solution of the Equation (3.76) was found by analogy to scalar solutions, which was given by Equation (3.25) and reproduced below:

$$\mathbf{x}(t) = e^{A(t - t_0)}\, \mathbf{x}(t_0) + \int_{t_0}^{t} e^{A(t-\tau)}\, B\mathbf{u}(\tau)\, d\tau \tag{3.78}$$

Equation (3.30) can be written as $\mathbf{x}(t) = \phi(t - t_0)\, \mathbf{x}\,(t_0) + \int_{t_0}^{t} \phi(t - \tau)\, B\, u(\tau)\, d\tau$ (3.79)

where $\phi(t) = e^{At}$ is known as the state transition matrix, t_0 is the initial time and $\mathbf{x}(t_0)$ is the initial state.

Chapter 3

The state transition matrix $\phi(t) = e^{At}$ represents a power series of the matrix At, for $t \geq 0$,

$$e^{At} = I + At + \frac{1}{2!}A^2 t^2 + \frac{1}{3!}A^3 t^3 + ... \tag{3.80}$$

where I is the identity matrix with dimension compatible to the system matrix A.

That the value of **x(t)** in Equation (3.78) is the solution of the state equation (3.76) can be easily verified by substituting its value in the Equation (3.76).

3.6.1 Properties of the State Transition Matrix

Some useful properties of the state transition matrix $\phi(t)$ are recorded below :

1. $\phi(0) = e^{A.0} = I$ (identity matrix) (3.81)
2. $\phi(t) = e^{At} = (e^{-At})^{-1} = [\phi(-t)^{-1}]$ or $\phi^{-1}(t) = \phi(-t)$ (3.82)
3. $\phi(t_1 + t_2) = e^{A(t_1+t_2)} = e^{At_1} . e^{At_2} = \phi(t_1)\, \phi(t_2) = \phi(t_2)\, \phi(t_1)$ (3.83)
4. $[\phi(t)]^n = \phi(nt)$ (3.84)
5. $\phi(t_1 - t_2)\, \phi(t_2 - t_3) = \phi(t_1 - t_3)$ for any t_1, t_2, t_3 (3.85)
6. $\dfrac{d\phi(t)}{dt} = A\phi(t)$ (3.86)

Example 3.7 Let us consider a single input single output system of Example 3.3 which is reproduced below for quick reference

$$A = \begin{bmatrix} 0 & 1 \\ -10 & -7 \end{bmatrix},\ b = \begin{bmatrix} 0 \\ 1 \end{bmatrix},\ c = [1\ 0] \text{ and } d = [0\ 0]$$

We are interested to find its solution with initial condition $x'(t_0) = x'(0) = [0 \quad 0]$ and unity step input $u(t) = u_s(t)$.

The resolvent matrix $\phi(s)$ given by relation (3.44) is written as :

$$\phi(s) = \begin{bmatrix} \dfrac{s+7}{(s+5)(s+2)} & \dfrac{1}{(s+5)(s+2)} \\ \dfrac{-10}{(s+5)(s+2)} & \dfrac{s}{(s+5)(s+2)} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{3}\left(\dfrac{5}{s+2} - \dfrac{2}{s+5}\right) & \dfrac{1}{3}\left(\dfrac{1}{s+2} - \dfrac{1}{s+5}\right) \\ \dfrac{10}{3}\left(\dfrac{1}{s+2} - \dfrac{1}{s+5}\right) & \dfrac{1}{3}\left(\dfrac{5}{s+5} - \dfrac{2}{s+2}\right) \end{bmatrix}$$

The state transition matrix $\phi(t)$ is obtained by taking Laplace inverse transform of $\phi(s)$ and is given by

$$\phi(t) = \begin{bmatrix} \dfrac{1}{3}(5e^{-2t} - 2e^{-5t}) & \dfrac{1}{3}(e^{-2t} - e^{-5t}) \\ \dfrac{10}{3}(e^{-2t} - e^{-5t}) & \dfrac{1}{3}(5e^{-5t} - 2e^{-2t}) \end{bmatrix} \tag{3.87}$$

Substituting the value of $x'(0) = [0\ 0]$ and unity step input $u_s(t)$ in Equation (3.79), we get

$$x(t) = \int_0^t \phi(t-\tau)\, bu(\tau)\, d\tau = \begin{bmatrix} \dfrac{1}{3}\displaystyle\int_0^t (5e^{-2(t-\tau)} - 2e^{-5(t-\tau)})\, d\tau & \dfrac{1}{3}\displaystyle\int_0^t (e^{-2(t-\tau)} - e^{-5(t-\tau)})\, d\tau \\ \dfrac{10}{3}\displaystyle\int_0^t (e^{-2(t-\tau)} - e^{-5(t-\tau)})\, d\tau & \dfrac{1}{3}\displaystyle\int_0^t (5e^{-5(t-\tau)} - 2e^{-2(t-\tau)})\, d\tau \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{1}{3}\int_0^t (e^{-2(t-\tau)} - e^{-5(t-\tau)})\, d\tau \\ \frac{1}{3}\int_0^t (5e^{-5(t-\tau)} - 2e^{-2(t-\tau)})\, d\tau \end{bmatrix} = \begin{bmatrix} \frac{1}{10} - \frac{1}{6}e^{-2t} + \frac{1}{15}e^{-5t} \\ \frac{1}{3}e^{-2t} - \frac{1}{3}e^{-5t} \end{bmatrix}$$

Therefore, $y(t)$ is computed as: $y(t) = cx(t) + du(t) = \frac{1}{10} - \frac{1}{6}e^{-2t} + \frac{1}{15}e^{-5t}$, $t \geq 0$.

REVIEW EXERCISE

RE3.1 Obtain the state variable representation for the following differential equation, with phase variables as state variables:

$$\frac{d^2y}{dt^2} + 2\frac{dy}{dt} + 26y = 26\, u(t)$$

Also find the resolvent and state transition matrix

Ans: $A = \begin{bmatrix} 0 & 1 \\ -26 & -2 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$, $c = [26\ 0]$

Resolvent matrix $\phi(s) = \frac{1}{\Delta}\begin{bmatrix} s+2 & 1 \\ -26 & s \end{bmatrix}$, where $\Delta = s^2 + 2s + 26$ and

$$\phi(t) = \frac{e^{-t}}{5}\begin{bmatrix} 5\cos 5t + \sin 5t & \sin 5t \\ -26 \sin 5t & 5\cos 5t - \sin 5t \end{bmatrix}$$

RE3.2 For the system matrix A given below

$$A = \begin{bmatrix} 0 & 6 \\ -1 & -5 \end{bmatrix}$$

(a) Find the state transition matrix $\phi(t)$.

Ans: $\phi(t) = \begin{bmatrix} 3e^{-2t} - 2e^{-3t} & 6e^{-2t} - 6e^{-3t} \\ e^{-3t} - e^{-2t} & 3e^{-3t} - 2e^{-2t} \end{bmatrix}$

RE3.3 For the differential equation given below, obtain the state variable representation in terms of phase variables and find the resolvent matrix.

$$\frac{d^3y}{dt^3} + 6\frac{d^2y}{dt^2} + 13\frac{dy}{dt} + 20y = 20u(t).$$

Ans: $A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -20 & -13 & -6 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$, $c = [20\ 0\ 0]$

and $\phi(s) = \frac{1}{\Delta}\begin{bmatrix} s^2 + 6s + 13 & s+6 & 1 \\ -20 & s(s+6) & s \\ -20s & -(13s+20) & s^2 \end{bmatrix}$, where $\Delta = s^3 + 6s^2 + 13s + 20$.

RE3.4 Obtain the state variable representation for the system in Fig. RE3.4(a).

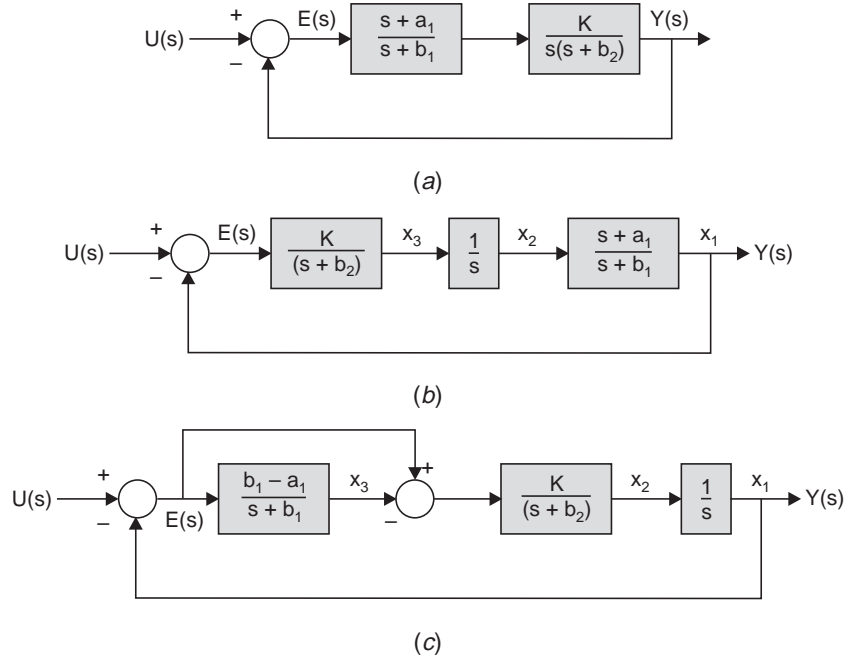



Fig. RE3.4 State variable representation

Hints: In order to avoid the derivative of the input u, because of the presence of the term $(s + a_1)$ in the numerator in the first block in Fig. RE3.4(a), use equivalent representations in part (b) or (c) in the figure.

Note that in drawing the Fig. RE3.4 (c), the factor $\dfrac{s+a_1}{s+b_1}$ has been recast as $1 - \dfrac{b_1 - a_1}{s + b_1}$

The state variable representations corresponding to Fig. RE3.4 (b) and (c) respectively, are given by :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -b_1 & a_1 & 1 \\ 0 & 0 & 1 \\ -K & 0 & -b_2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ K \end{bmatrix} u \; ; \quad y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ -K & -b_2 & -K \\ -(b_1 - a_1) & 0 & -b_1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ K \\ (b_1 - a_1) \end{bmatrix} u \; ; \quad y = [1 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

RE3.5 Consider the case of the two masses connected as shown in Fig. RE3.5, where $k_1$and k_2 are spring constants and β_1 and β_2 are the sliding friction constants of masses m_1 and m_2 respectively. Find the state variable matrix differential equation for the system.

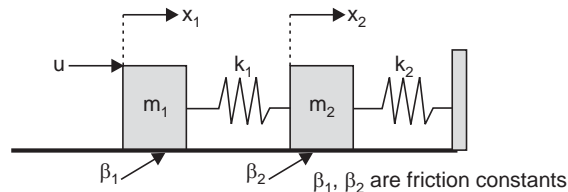



Fig. RE3.5

Hints : The differential equation is given by:

$u = m_1\ddot{x}_1 + \beta_1\dot{x}_1 + k_1x_1 - k_1x_2$ and $0 = -k_1x_1 + m_2\ddot{x}_2 + \beta_2\dot{x}_2 + k_2x_2 + k_1x_2$

PROBLEMS

P3.1 For the system matrix $A = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix}$

(*a*) find the state transition matrix $\phi(t)$ and (*b*) find $x(t)$ for the initial conditions $x_1(0) = x_2(0) = 1$.

P3.2 Obtain the state variable representation with phase variables as state variable for the system described by the transfer function

$$G(s) = \frac{Y(s)}{U(s)} = \frac{5(s+10)}{(s+4)(s+1)}$$

P3.3 Obtain the state-transition matrix $\phi(t)$ and its inverse, $\phi^{-1}(t)$ of the following system :

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -2 & 0 \\ 0 & -5 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix};$$

Ans: $\phi(t) = e^{At} = £^{-1}[sI - A)^{-1}] = \begin{bmatrix} e^{-2t} & 0 \\ 0 & e^{-5t} \end{bmatrix}$

$$\phi^{-1}(t) = \phi(-t) = \begin{bmatrix} e^{2t} & 0 \\ 0 & e^{5t} \end{bmatrix}$$

P3.4 Compute $\phi(s)$ for a system with

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$$

Ans: $\phi(s) = \frac{1}{\Delta}\begin{bmatrix} s^2 + a_1s + a_2 & s + a_1 & 1 \\ -a_3 & s(s+a_1) & s \\ -a_3s & -(a_2s + a_3) & s^2 \end{bmatrix}$, where $\Delta = s^3 + a_1s^2 + a_2s + a_3$

P3.5 Show that the transfer function of the lag-lead network (vide Chapter 8) in Fig. P3.5 is given by:

$$\frac{E_0(s)}{E_i(s)} = \frac{Z_2}{Z_1 + Z_2} = \frac{(R_1C_1s+1)(R_2C_2s+1)}{(R_1C_1s+1)(R_2C_2s+1) + R_1C_2s} = \frac{(sT_1+1)(sT_2+1)}{\left(s\frac{T_1}{\alpha}+1\right)(s\alpha T_1 + 1)}$$

$$= \frac{(s+a_1)(s+a_2)}{(s+\alpha a_1)\left(s+\frac{a_2}{\alpha}\right)}$$, where $Z_1 = \frac{R_1}{(R_1C_1s+1)}$, $Z_2 = R_2 + \frac{1}{C_2s}$, $T_1 = R_1C_1 = 1/a_1$,

$T_2 = R_2C_2 = 1/a_2$, and $R_1C_1 + R_2C_2 + R_1C_2 = \frac{T_1}{\alpha} + \alpha T_2$, $\alpha > 1$

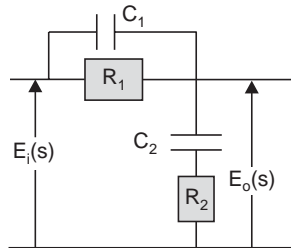



Fig. P3.5 Transfer function of RC lag-lead network

Obtain the state variable representation from the transfer function with phase variables as state variables with R_1 = 1.2 MΩ, R_2 = 10 MΩ, C_1 =1.0 micro Farad, and C_2 = 0.1 micro Farad.

P3.6 A single-input, single-output system is represented by the state variable equations:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -7 & -10 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u \; ; y = [\,10\ 0\,]\, x.$$

Determine the transfer function G(*s*) = Y(*s*)/U(*s*) using the (*a*) signal-flow graph and (*b*) using relations (3.44) and (3.46) in the text

Ans: $$G(s) = \frac{10}{s^2 + 7s + 10}$$

P3.7 (*a*) Consider the circuit shown in Fig. P3.7. Show that the transfer function from V_i to V_o is given by

$$V_o(s)/V_i(s) = \frac{1}{R_1R_2C_1C_2s^2 + (R_1C_1 + R_2C_2 + R_1C_2)s + 1}$$

(*b*) With R_1 = 5MΩ, R_2 = 10 MΩ, C1 = 0.5 μF, and C_2 = 0.1μF, show that a state variable representation is given by:

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} v_i$$

(c) Using the state variable representation from part (*b*), plot the unit step response with a step input voltage.

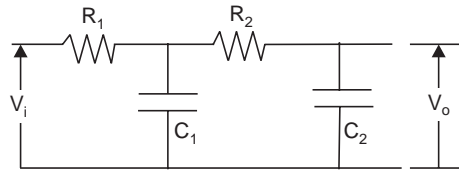



Fig. P3.7

P3.8 The dynamics of temperature variation in a building with solar collector room-heating system is described by

$$\frac{dx_1}{dt} = 3x_1 + u_1 + u_2$$

$$\frac{dx_2}{dt} = 2x_2 + u_2 + d$$

where x_1 = temperature deviation from desired equilibrium, and x_2 = temperature of the storage material like water in a tank. The control inputs u_1 and u_2 are the flow rates of convectional and solar heat, respectively caused by forced air. The disturbance on solar storage temperature caused by overcast skies is represented by *d*. Write the matrix equations and solve for the system response when $u_1 = 0$, $u_2 = 1$, and $d = 1$, with zero initial conditions.

P3.9 Extenders are robot manipulators that help increase the strength of the human arm in load maneuvering tasks. The transfer function of an extender is represented by :

$$\frac{Y(s)}{U(s)} = G(s) = \frac{40}{s^2 + 4.8s + 4}$$

where U(*s*) is the Laplace transform of the force of the human hand applied to the robot manipulator, and Y(*s*) is the Laplace transform of the force of the robot manipulator applied to the load. Using the phase variables as state variables obtain the state variable representation and the state transition matrix for the system.

P3.10 An RLC circuit is shown in Fig. P3.10 (*a*). Make a choice of suitable set of state variables and obtain the state variable representation. (*b*) Draw the signal flow graph in the s-domain assuming that the initial conditions of the states are non-zero.

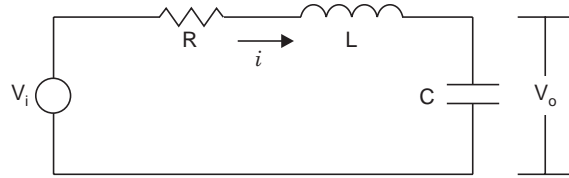



Fig. P3.10 RLC series circuit

P3.11 Fluid control systems are reliable over a wide range of temperature and insensitive to electromagnetic and nuclear radiation, acceleration, and vibration. A closed loop speed control for steam turbine is shown in Fig. P3.11 that employs hydraulic components. A tuning fork is used to provide reference speed for the steam turbine with an angular speed of 10,000 rpm. The amplification within the system is achieved by using a fluid jet deflection amplifier. The block diagram of the system is shown in Fig. P3.11 with typical normalized values $\beta = 0.1$, J = 2 and $k_1 = 0.5$.

(*a*) Determine the closed-loop transfer function

$$G(s) = \frac{\omega(s)}{R(s)}$$

and obtain the state variable representation and determine the characteristic equation obtained from the system matrix A.

(*b*) Find the transfer function $G_d(s) = \dfrac{\omega(s)}{T_d(s)}$

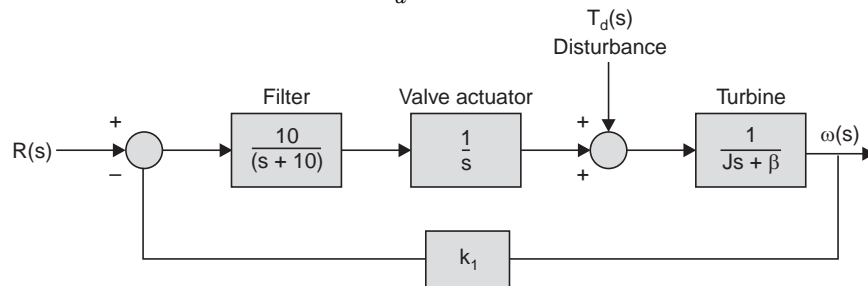



Fig. P3.11 Steam turbine control

P3.12 A two-axis control system is shown in Fig. P3.12 where a set of state variables is indicated. The gain of each axis is A_1 and A_2, respectively. (*a*) Obtain the state variable representation. (*b*) Find the characteristic equation from the A matrix. (*c*) Determine the state transition matrix for $A_1 = 1$ and $A_2 = 2$.

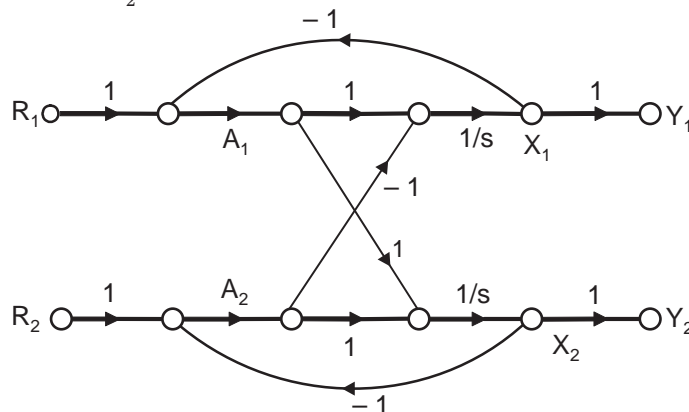



Fig. P3.12 Coupled two-axis control system

P3.13 Determine the signal-flow graph and the state variable representation with (*a*) the phase variables as state variables and (*b*) the canonical (diagonal) variables as state variable when the transfer function is given by:

$$G(s) = \frac{Y(s)}{U(s)} \quad \frac{5s + 8}{s^3 + 7s^2 + 14s + 8}$$

P3.14 The state variable representation of a system is given by :

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix} x + \begin{bmatrix} 0 \\ b_1 \end{bmatrix} u(t),\ y = 8x_1.$$

(*a*) Determine the parameters, a_1, a_2, and b_1 such that it can be expressed in the form :

$$\dot{x} = \begin{bmatrix} -4 & 0 \\ 0 & -2 \end{bmatrix} x + \begin{bmatrix} 1 \\ 1 \end{bmatrix} u\ , \quad y = [-2 \ \ 2]\ x$$

Ans: $b_1 = 2$, $a_1 = 8$, $a_2 = 6$

P3.15 A system has a block diagram as shown in Fig. P3.15. Determine the state variable representation and the state transition matrix $\phi(t)$.

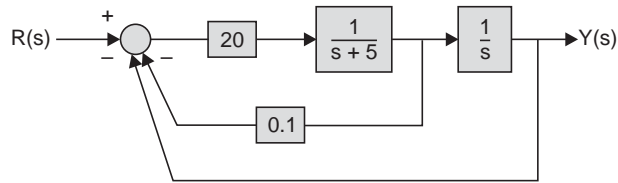



Fig. P3.15

P3.16 A closed-loop control system is shown in Fig. P3.16. (*a*) Determine the closed-loop transfer function $G_p(s) = Y(s)/U(s)$. (*b*) Find the state variable representation with phase variables as state variables. (*c*) Also obtain the state variable representation with physical variables as state variables as shown in Fig. P3.16.

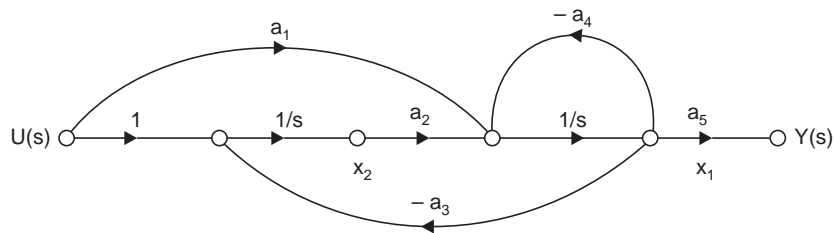



Fig. 3.16 Signal flow graph

CHAPTER

4

Analysis of Linear Systems

4.1 TIME-DOMAIN PERFORMANCE OF CONTROL SYSTEMS

After preparing the mathematical model of a system, the next important task of an engineer, as was mentioned in Section 1.7 of Chapter 1, is the analysis of the system in its working environment to asses its performance by using available mathematical tools. Linear systems may be analyzed in time domain as well as in frequency domain. However, the exact nature of the various inputs to which a control system is subjected to in an actual working environment, is not always known before hand. So, the performance of a control system is evaluated in the laboratory by using some typical input signals, especially for comparing alternative controller designs and verifying design concepts. In the design stage, performance criteria are specified with respect to these test signals and the control systems are designed to meet these criteria. The most commonly used test input signals are: *Step, Ramp, Parabolic, Impulse* and *Sinusoidal functions*. Since, these inputs are simple functions of time, the performance analysis of the systems are rather easy to perform. A control system designed on the basis of a test signals, has been found to perform satisfactorily when subjected to actual inputs in its normal working environment.

The step function is very useful as a test signal since its initial instantaneous jump in amplitude reveals a great deal about a system's performance with respect to inputs with abrupt changes. Besides, as the step function contains, in principle, a wide range of frequencies in its spectrum, it is equivalent to the application of numerous sinusoidal signals with a wide range of frequencies. If the system is likely to be subjected to shock inputs, then the impulse function may be a good choice as a test signal.

The ramp function is another important test signal that is used to test the systems' ability to follow velocity type inputs. A parabolic function is one degree faster than that of the ramp. One can define inputs faster than the ramp function, but, it is never used in practice. This is because of the fact that, high-order integrators in the forward path needed to track the high-order inputs accurately will make the closed loop system susceptible to instability.

A sinusoidal input function should be chosen as a test signal if the system is very often subjected to a slow but periodic input signal.

4.2 TYPICAL TEST INPUTS

4.2.1 The Step-Function Input

The characteristic of step function input, along with its Laplace transform has already been discussed in Section 2.3 of Chapter 2. The plot of the step input of strength R is shown in

Fig. 4.1(*a*) and the Laplace transform of the step input of strength R is given in Equation (2.3) and is reproduced below:

$$R(s) = R/s \tag{4.1a}$$

4.2.2 The Ramp-Function Input

The ramp function and its Laplace transform have also been presented in Section 2.3 of Chapter 2. The plot of ramp function with slope R is shown in Fig. 4.1(b) and its Laplace transform is given in Equation (2.5) and is reproduced below:

$$R(s) = R/s^2 \tag{4.1b}$$

4.2.3 The Impulse-Function Input

A rectangular pulse is an engineering approximation of the impulse function. Some important properties of the impulse function are presented in Section 2.3 of Chapter 2. It has been mentioned that an impulse function of strength A/T and duration T is defined as

$$\begin{aligned} f(t) &= \lim_{T \to 0} \frac{A}{T} \quad \text{for } 0 < t < T \\ &= 0 \qquad\quad \text{for } t < 0,\ t > T \end{aligned} \tag{4.1c}$$

The strength of an impulse is represented by its area. The strength of the impulse represented by the relation (4.1*c*) is A, since the height of the impulse function is A/T and the duration is T. The Laplace transform of an impulse function of strength A is given by [see Equation (2.11) in Chapter 2].

$$\pounds\, f(t) = A \tag{4.1d}$$

A plot of a train of impulse functions of strength $f(0)$ $f(T)$ $f(2T)$ are shown in Fig. 4.1(*c*).

4.2.4 The Parabolic-Function Input

A parabolic function may be defined as:

$$r(t) = \frac{1}{2} Rt^2 u_s(t) \tag{4.2a}$$

where R is a real constant and $u_s(t)$ is a unity step function. The graphical representation of the parabolic function is shown in Fig. 4.1(*d*)

The Laplace transform of the *Parabolic-Function* in Equation (4.2*a*) can be written from the 4th row of Laplace transform Table 2.1 of Chapter 2 with $n = 3$ as :

$$R(s) = R/s^3 \tag{4.2b}$$

All these test signals are easy to represent in time domain as well as in frequency domain. However, considering the functions step, ramp and parabolic, they become progressively faster with respect to time.

(a)

(b)

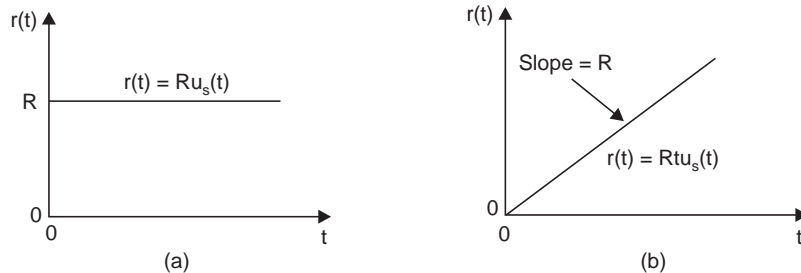
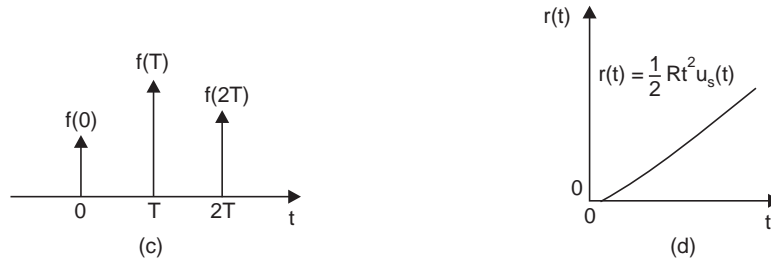



Fig. 4.1 Basic time-domain test signals (a) Step function (b) Ramp function (c) Impulse function (d) Parabolic function

4.3 TRANSIENT STATE AND STEADY STATE RESPONSE OF ANALOG CONTROL SYSTEM

When a system is subjected to a test input, the response in time may be divided in two parts. The first part of the response immediately after the application of the input is called *transient response* whereas the response long after the application of the input, as $t \to \infty$ is known as *steady state* response. In stable control systems, the *transient response* dies as time becomes large compared to the smallest time constant of the system and *the steady state* response begins. The time response of an analog control system, denoted by $y(t)$, can, in general, be written in terms of its two components as:

$$y(t) = y_t(t) + y_{ss}(t) \tag{4.3}$$

where $y_t(t)$ denotes the *transient response*, and $y_{ss}(t)$ denotes the *steady-state response*.

Both the steady state and the transient response are important for a control system. For example, consider the case of controlling the position of an anti aircraft Gun Turret. A schematic diagram for controlling gun turret in one plane is shown in Fig. 4.2. The very nature of application demands that the gun is aligned to the enemy target in minimum possible time following a step input. The analysis will show that such systems should have a fast rise time in the transient response even though it may be accompanied by large overshoot and undershoot (vide Fig. 4.3).

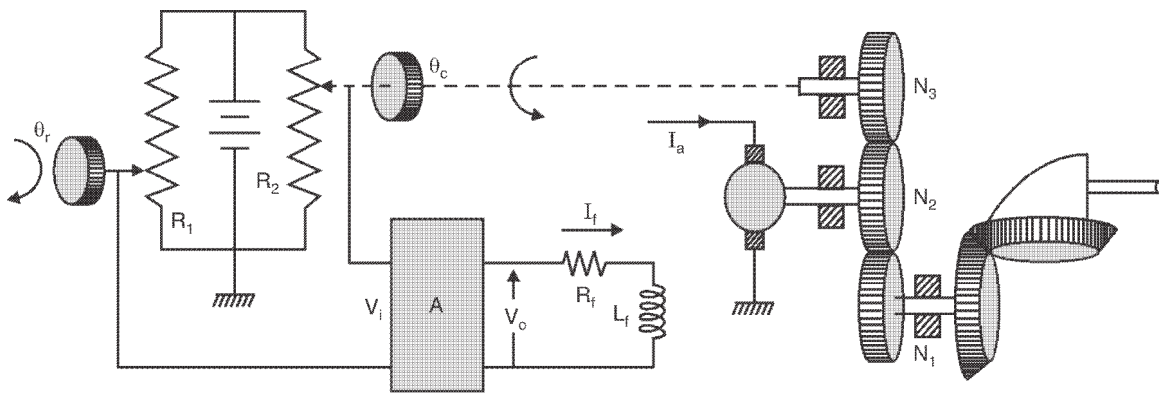



Fig. 4.2 Schematic diagram for controlling the position of a gun turret

Now consider another servo position control system, which is used for controlling the position of hull of a big ship, where the large overshoot and undershoot cannot be tolerated but a slower response is acceptable if the final steady state error is small. Even though, both the

control systems are used for controlling the angular position of mechanical loads, the transient response is of paramount importance in one in contrast to the zero steady state error in the other.

If the steady-state response of the output does not agree exactly with the input, the system is said to have a finite *steady-state error*.

The study of a control system in the time domain essentially involves the evaluation of the transient and the steady-state responses of the system. In the design stage of a controller, specifications are usually given in terms of the transient and the steady-state performance, which are to be satisfied by the closed loop system.

4.4 PERFORMANCE SPECIFICATION OF LINEAR SYSTEMS IN TIME-DOMAIN

4.4.1 Transient Response Specifications

Fig. 4.3 illustrates a typical unit-step response of a linear analog control system. With reference to the unit-step response, performance criteria commonly used for the characterization of linear control systems in the time domain are defined as follows :

1. *Maximum Percent Overshoot.* Let y_{max} denote the maximum value of the unit-step response $y(t)$, and y_{ss} be the steady-state value of $y(t)$, and $y_{max} \geq y_{ss}$. The maximum percent overshoot of $y(t)$ is defined as

$$\textit{Maximum percent overshoot} = \frac{y_{max} - y_{ss}}{y_{ss}} \times 100\% \tag{4.4}$$

The maximum overshoot is often used as a measure of the relative stability of a control system. A system with a large overshoot is usually not desirable in many applications. For design purposes, the maximum overshoot is often given as a time-domain specification. The unit-step response illustrated in Fig. 4.3 shows that the maximum overshoot occurs at the first overshoot, which, however, may not be the case for all types of control systems.

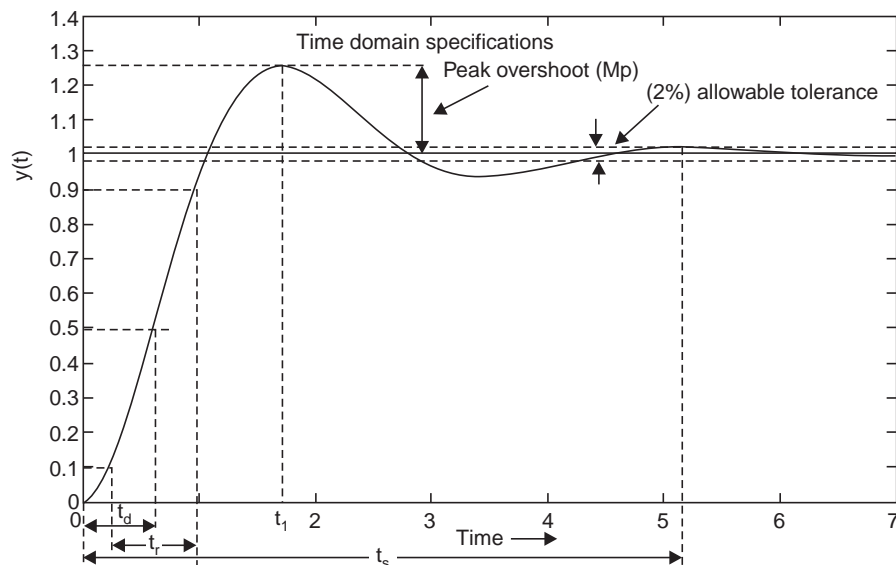



Fig. 4.3 Typical response of a second order linear system with step input

Delay Time: The delay time t_d is normally defined as the time required for the step response to reach 50 percent of its final value (vide Fig. 4.3).

2. *Rise Time*. The rise time t_r is defined as the time required for the step response to rise from 10 to 90 percent of its final value (vide Fig. 4.3). An alternative measure is to represent the rise time as the reciprocal of the slope of the step response at the instant that the response is equal to 50 percent of its final value.

3. *Settling Time.* The settling time t_s is defined as the time required for the step response to decrease and stay within a specified percentage of its final value. A frequently used figure is 5 percent and 2 percent.

The four quantities just defined give a direct measure of the transient characteristics of a control system in terms of the unit-step response. These time-domain specifications are relatively easy to measure when the step response is well defined, as shown in Fig. 4.3. Analytically, these quantities are difficult to establish, except for simple systems of order lower than third.

4.5 TRANSIENT RESPONSE OF A PROTOTYPE SECOND-ORDER SYSTEM

We shall undertake here the analysis of a second-order system that will help to form a basis for the understanding of analysis and design of higher-order systems.

Consider that a second-order control system is represented by the block diagram shown in Fig. 4.4. The open-loop transfer function of the system is

$$G(s) = \frac{Y(s)}{E(s)} = \frac{\omega_n^2}{s(s + 2\delta\,\omega_n)} \tag{4.5}$$

where δ and ω_n are real constants and are referred to as *damping ratio* and *undamped natural frequency,* respectively, of a second order system . The closed-loop transfer function of the system with $H(s) = 1$ is

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\,\delta\,\omega_n s + \omega_n^2} \tag{4.6}$$

The system in Fig. 4.4 with the transfer functions given by Equations (4.5) and (4.6) is defined as the **prototype second-order system.**

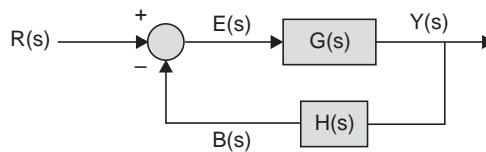



Fig. 4.4 A prototype second-order control system

The characteristic equation of the closed-loop system is obtained by setting the denominator of Equation (4.6) to zero.

$$\Delta(s) = s^2 + 2\delta\omega_n s + \omega_n^{\,2} = 0 \tag{4.7}$$

The roots of the characteristic equation are given by

$$s_{1,\,2} = -\,\delta\omega_n \pm \omega_n\sqrt{\delta^2 - 1} \tag{4.8}$$

The dynamic behavior of the second-order system can, therefore, be described in terms of two parameters of the roots of the characteristic equation, δ and ω_n, respectively. Let us now plot the variation of the roots in Equation (4.8) as the parameters δ and ω_n are varied.

4.5.1 Locus of Roots for the Second Order Prototype System

4.5.1.1 Constant ω_n Locus

From Equation (4.8), it is apparent that, with ω_n held constant, if δ is varied from 0 to $+\infty$, the two roots of the characteristic equation will originate from $\pm j\omega_n$ corresponding to $\delta = 0$, move along the arc of a semicircle of radius ω_n and meet at $-\omega_n$ on the real axis corresponding to $\delta = 1$. When δ is further increased, the two branches move away from each other in opposite directions along the real axis-one branch terminates at $s = 0$ and the other terminates at $s = -\infty$ as δ approaches $+\infty$ [see Fig. 4.5(*a*)].

4.5.1.2 Constant Damping Ratio Line

The complex conjugate roots s_1, s_2 are shown in Fig. 4.5(*b*) for the case $0 < \delta < 1$. When the root s_1 is joined to the origin of the *s*-plane by a straight line, it makes an angle θ with negative real axis. It can be shown with the help of Equation (4.8) and the property of triangles, that $\cos \theta = \delta$. Therefore, if δ is held constant and ω_n is varied, the roots will move on straight lines making an angle $\theta = \cos^{-1}(\delta)$ as shown in Fig. 4.5(*b*). It will be found [see Equation (4.14)] that when the damping δ is constant, the percent overshoot to step input response will remain constant but the settling time will be different with different ω_n [see Fig. 4.6(*a*)].

4.5.1.3 Constant Settling Time

A glance at the Equation (4.8) reveals that the real part of the roots can be kept constant when $\delta\,\omega_n$ is maintained to a constant value with various combinations of δ and ω_n. The locus

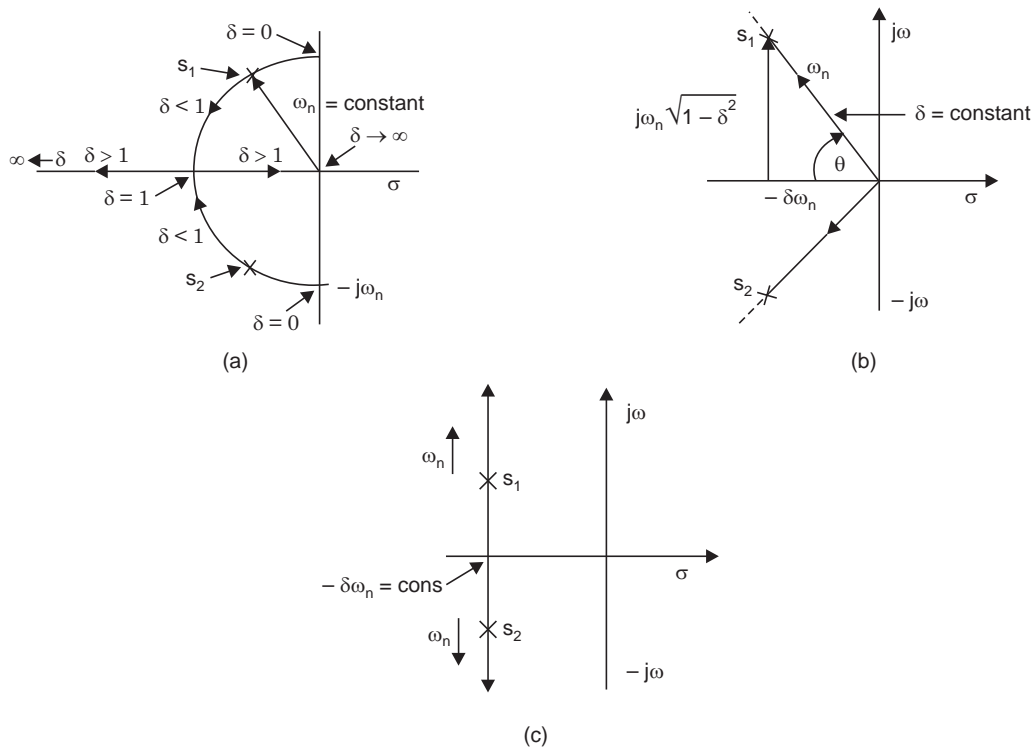



Fig. 4.5 Locus of roots of characteristic equation of a prototype second order system when damping ratio and natural frequency is varied. (a) ω_n is constant and δ is varied from 0 to $+\infty$, (b) δ is constant and $\omega_n > 0$ is varied (c) when both δ and ω_n are varied keeping their product constant

of roots will appear as shown in Fig. 4.5(c). A constant value of $\delta\omega_n$ will be found to produce a constant settling time [vide Equation (4.16)] in the transient response to step input, but will give rise to different values of percent overshoots as depicted in Fig. 4.6(b).

4.5.2 Transient Response with Constant ω_n and Variable δ

From the point of view of transient response, four distinct cases are of interest to us when ω_n is held constant and the damping ratio δ takes on different values.

1. *Underdamped case* : $0 < \delta < 1$

If $0 < \delta < 1$, the closed-loop poles are given by

$$s_{1,\,2} = -\,\delta\,\omega_n \pm j\omega_n\sqrt{1-\delta^2}$$

which are complex conjugates and lie on the semicircle in the left-half s-plane as shown in Fig. 4.5(a).

The system is then referred to as underdamped, and the transient response will be found to be oscillatory [see Fig. 4.7].

2. *Critically damped case* : $\delta = 1$

If $\delta = 1$, the system is called critically damped. In this case, both the roots are equal and are located at $s_{1,2} = -\,\omega_n$ as indicated in Fig. 4.5(a). The step response is monotonic without any overshoots and undershoots.

3. *Overdamped case* : $\delta > 1$

If $\delta > 1$, both the roots are lying on real axis and are given by

$$s_1 = -\,\delta\omega_n + \omega_n\sqrt{\delta^2-1} \text{ and } s_2 = -\,\delta\omega_n - \omega_n\sqrt{\delta^2-1}.$$

The location of the poles in the s-plane are indicated in Fig. 4.5(a). The transient response of over damped systems is also monotonic without any oscillation [see Fig. 4.7].

4. *Undamped case* : $\delta = 0$

In the undamped case, the poles are located on the imaginary axis at $s_1 = +\,j\omega_n$ and $s_2 = -\,j\omega_n$ as indicated in Fig. 4.5(a). The transient response is oscillatory in nature and does not die out [vide Fig. 4.7].

These observations are applicable to the characteristic equation of prototype second order system with closed loop system transfer function given by Equation (4.6). However, the dynamics of a higher order system with a dominant pair of complex conjugate pole pairs are approximated by a second order model to exploit the above facts as a starting point of controller design by classical techniques (vide Chapters 7 and 8).

4.5.2.1 Step Input Response

We shall now solve for the response of the system shown in Fig. 4.4 subjected to a unit-step input. We shall consider three different cases; the underdamped ($0 < \delta < 1$), critically damped ($\delta = 1$), and overdamped ($\delta > 1$) cases.

(i) **Underdamped case ($0 < \delta < 1$) :**

In this case, $Y(s)/R(s)$ can be written as

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{(s+\delta\omega_n + j\omega_d)(s+\delta\omega_n - j\omega_d)}$$

where $\omega_d = \omega_n\sqrt{1-\delta^2}$. The frequency ω_d is called the *damped natural frequency*. For a unit-step input, R(s) = 1/s, so, Y(s) can be written as

$$Y(s) = \frac{\omega_n^2}{(s + 2\delta\omega_n s + \omega_n^2)\, s} \tag{4.9a}$$

The inverse Laplace transform of Equation (4.9a) can be obtained easily if Y(s) is written in the following form:

$$\begin{aligned} Y(s) &= \frac{1}{s} - \frac{s + 2\delta\omega_n}{s^2 + 2\delta\omega_n s + \omega_n^2} \\ &= \frac{1}{s} - \frac{s + \delta\omega_n}{(s + \delta\omega_n)^2 + \omega_d^2} - \frac{\delta\omega_n}{(s + \delta\omega_n)^2 + \omega_d^2} \\ &= \frac{1}{s} - \frac{s + \delta\omega_n}{(s + \delta\omega_n)^2 + \omega_d^2} - \frac{\delta}{\sqrt{1-\delta^2}}\frac{\omega_d}{(s + \delta\omega_n)^2 + \omega_d^2} \end{aligned} \tag{4.9b}$$

From row 21 and 20 in Laplace transform Table 2.1 of Chapter 2, we find respectively

$$£^{-1}\left[\frac{s + \delta\omega_n}{(s + \delta\omega_n)^2 + \omega_d^2}\right] = e^{-\delta\omega_n t} \cos \omega_d t$$

$$£^{-1}\left[\frac{\omega_d}{(s + \delta\omega_n)^2 + \omega_d^2}\right] = e^{-\delta\omega_n t} \sin \omega_d t$$

Hence the inverse Laplace transform of Equation (4.9b) is obtained as

$$£^{-1}\,[Y(s)] = y(t) = 1 - e^{-\delta\omega_n t}\left(\cos\omega_d t + \frac{\delta}{\sqrt{1-\delta^2}}\sin\omega_d t\right)$$

(a)

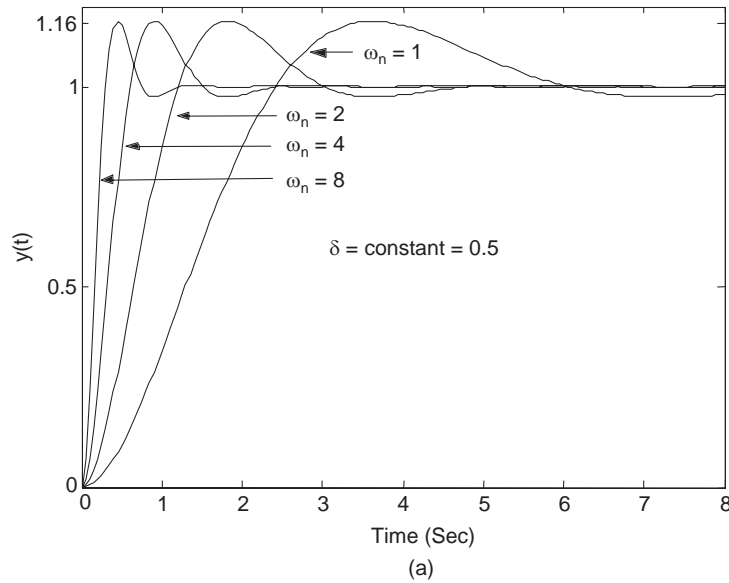
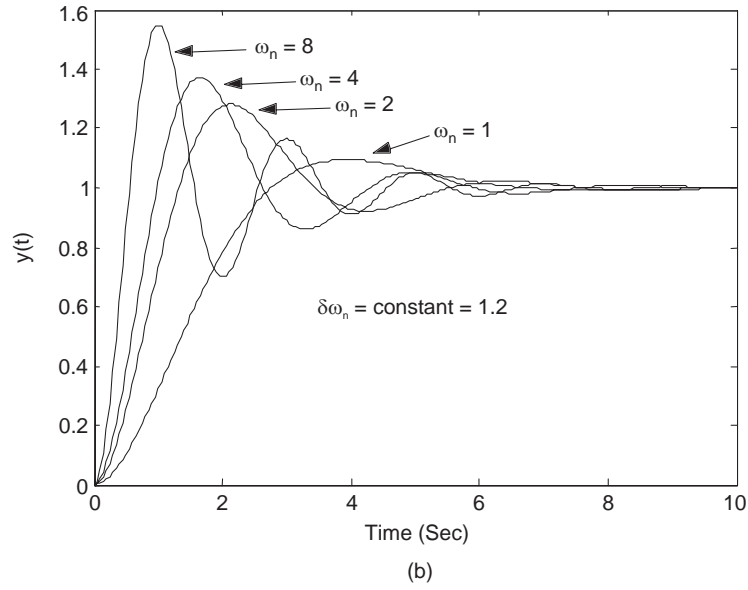



Fig. 4.6 Step response of a prototype second order system (a) with constant damping ratio and varying ω_n(b) both δ and ω_n are varied keeping their product, $\delta\omega_n$ constant

$$= 1 - \frac{e^{-\delta\omega_n t}}{\sqrt{1-\delta^2}} \sin(\omega_d t + \theta), t \geq 0 \tag{4.10}$$

where $\omega_d = \omega_n \sqrt{1-\delta^2}$ and $\theta = \tan^{-1} \frac{\sqrt{1-\delta^2}}{\delta} = \cos^{-1} \delta$

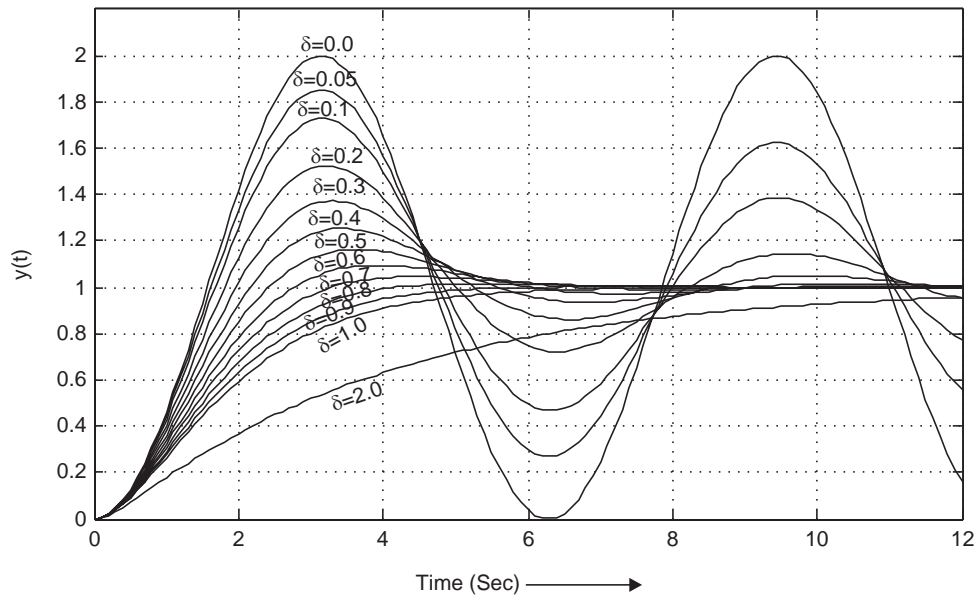



Fig. 4.7 Step response of a prototype second order system with constant ω_n and variable δ

This result can also be obtained directly by using row 24 in Laplace transforms Table 2.1 of Chapter 2. From Equation (4.10), it can be seen that the frequency of transient oscillation is the damped natural frequency ω_d and it varies with the damping ratio δ.

If the damping ratio δ is equal to zero, the response becomes undamped and oscillations continue indefinitely. The response $y(t)$ for the zero damping case may be obtained by substituting $\delta = 0$ in Equation (4.10), yielding

$$y(t) = 1 - \cos \omega_n t,\ t \geq 0 \tag{4.11}$$

Thus, we find from Equation (4.11) that ω_n represents the undamped natural frequency of the system. That is, ω_n is that frequency at which the system would oscillate if the damping were decreased to zero. If the linear system has any amount of damping, the undamped natural frequency is ω_d, which is equal to $\omega_n\sqrt{1-\delta^2}$. This frequency is always lower than the undamped natural frequency. An increase in δ would reduce the damped natural frequency ω_d. If δ is made greater than unity, the response becomes overdamped and will not oscillate.

(*ii*) **Critically damped case** ($\delta = 1$)**:** If the two poles of Y(*s*)/R(*s*) are nearly equal, the system may be approximated by a critically damped one.

For a unit-step input, R(*s*) = 1/*s* and hence Y(*s*) can be written as

$$Y(s) = \frac{\omega_n^2}{(s+\omega_n)^2 s} \tag{4.12}$$

The inverse Laplace transform of Equation (4.12) may be found, with the help of row 18 in Table 2.1, as

$$y(t) = 1 - (1 + \omega_n t)e^{-\omega_n t},\ t \geq 0$$

(*iii*) **Overdamped case** ($\delta > 1$) **:** In this case, the two poles of Y(*s*)/R(*s*) are negative real and unequal. For a unit-step input, R(*s*) = 1/*s*. So, Y(*s*) can be written as

$$Y(s) = \frac{\omega_n^2}{(s-\alpha_1)(s-\alpha_2)s} \tag{4.13}$$

where $\alpha_1 = -\delta\omega_n + \omega_n\sqrt{\delta^2 - 1}$ and $\alpha_2 = -\delta\omega_n - \omega_n\sqrt{\delta^2 - 1}$

The inverse Laplace transform of Equation (4.13) can be found with the help of row 17 in Table 2.1, as

$$y(t) = 1 - \frac{\omega_n}{2\sqrt{\delta^2-1}}\left(\frac{e^{-\alpha_1 t}}{\alpha_1} - \frac{e^{-\alpha_2 t}}{\alpha_2}\right),\ t \geq 0$$

Thus, the response $y(t)$ includes two decaying exponential terms.

When δ is appreciably greater than unity, one of the two decaying exponentials decreases much faster than the other, so that the faster decaying exponential term (which corresponds to a smaller time constant) may be neglected.

Peak time t_p: Referring to Equation (4.10), we may obtain the peak time t_p by differentiating $y(t)$ with respect to time and letting this derivative equal to zero :

$$\left.\frac{dy}{dt}\right|_{t=t_p} = \frac{\delta\omega_n}{\sqrt{1-\delta^2}}e^{-\delta\omega_n t_p}\sin(\omega_d t_p + \theta) - \frac{\omega_d}{\sqrt{1-\delta^2}}e^{-\delta\omega_n t_p}\cos(\omega_d t_p + \theta) = 0$$

This yields the equation : $\tan(\omega_d t_p + \theta) = \dfrac{\sqrt{1-\delta^2}}{\delta} = \tan\theta$

Therefore, $\omega_d t_p = n\pi$, $n = 0, 1, 2.....$; where $\omega_d = \omega_n\sqrt{1-\delta^2}$

Since the peak time corresponds to the first peak overshoot, $\omega_d t_p = \pi$.

Hence $$t_1 = t_p = \frac{\pi}{\omega_d} = \frac{\pi}{\omega_n\sqrt{1-\delta^2}}$$

The peak time t_p corresponds to one-half cycle of the frequency of damped oscillation.

Maximum overshoot M_p: The maximum overshoot occurs at the peak time at $t = t_p = \pi/\omega_d$. Thus, from Equation (4.10) , M_p is obtained as

$$M_p = y(t_p) - y_{ss} = y(t_p) - 1$$

$$= -e^{-\delta\omega_n(\pi/\omega_d)}\left(\cos\pi + \frac{\delta}{\sqrt{1-\delta^2}}\sin\pi\right)$$

$$= e^{-(\delta\omega_n/\omega_d)\pi} = e^{-\pi(\delta/\sqrt{1-\delta^2})}$$

So, using the Equation (4.4) for maximum percent overshoot, it is found for the prototype second order system as

$$\textit{Maximum per cent overshoot} = 100e^{-\frac{\pi\delta}{\sqrt{1-\delta^2}}} \tag{4.14}$$

Settling time: The exact analytical expression for the settling time even for a simple second order system cannot be derived. However, it can be approximated for damping ratios in the range $0 < \delta < 0.7$ by the envelope of the damped sinusoidal output $y(t)$ as shown in Fig. 4.8. We can write the envelope from Equation (4.10) by setting the sin (.) term to unity and equating this envelope at the settling time $t = t_s$ to the limits of 2 percent band around the steady state output to get

$$1 \pm \frac{e^{-\delta\omega_n t_s}}{\sqrt{1-\delta^2}} = 1 \pm 0.02 \tag{4.15}$$

Therefore, $\omega_n t_s$ can be solved from the above relation as

$$\omega_n t_s = -\frac{1}{\delta}\ln\left(0.02\sqrt{1-\delta^2}\right)$$

Hence, the settling time t_s is found to be $t_s = \dfrac{1}{\delta\omega_n}\ln\left(0.02\sqrt{1-\delta^2}\right)$ (4.16)

The quantity $\ln(0.02\sqrt{1-\delta^2})$ varies from 3.91 to 4.42 as δ is varied from 0.01 to 0.7. So we have

$$t_s \le \frac{4.2}{\delta\omega_n} \approx \frac{4}{\delta\omega_n}, \text{ for } 0 < \delta < 0.7 \tag{4.17}$$

For a 5 percent band around the steady state output, the settling time t_s is approximately given by

$$t_s \le \frac{3.4}{\delta\omega_n} \approx \frac{3.12}{\delta\omega_n}, \text{for } 0 < \delta < 0.7 \tag{4.18}$$

Even for higher order systems, whose dynamics is governed by a pole pair of dominant roots, very often, the designers take advantage of the relations (4.17) and (4.18) for quick estimation of the settling time.

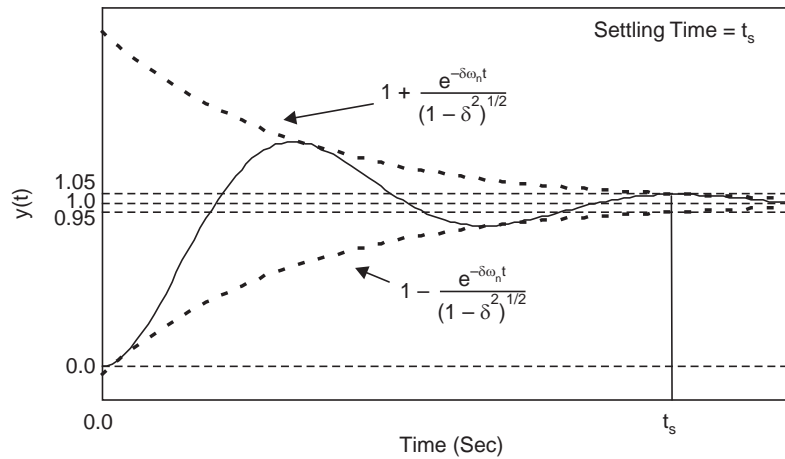



Fig. 4.8 Estimation of settling time of a second order system

4.6 IMPULSE RESPONSE OF A TRANSFER FUNCTION

Consider an open loop transfer function of a system given by :

$$G(s) = Y(s)/R(s) \tag{4.19a}$$

where R(s) is the Laplace transform of $r(t)$.

Now if the input $r(t)$ is an impulse of unity strength, $r(t) = 1.\delta(t)$, then its Laplace transform is given by :

$$R(s) = 1$$

So, we can write from Equation (4.19a),

$$Y(s) = G(s)$$

The impulse response $y(t)$ is obtained by taking the Laplace inverse of the transfer function G(s), written as $g(t)$.

That is, $$y(t) = g(t) \tag{4.19b}$$

So, we note that the response of a system in time domain stimulated by a unity impulse, reveals the characteristic signature of the system. In fact, the impulse response is used in real time identification of aircraft dynamics in some adaptive controller scheme, where a rectangular pulse with very small duration compared to the smallest time constant of the aircraft dynamics is applied periodically. The number of zero crossings in the impulse response in a fixed time interval is a measure of the damping ratio.

The unity impulse response of the closed loop prototype system with transfer function given in Equation (4.6) can be found by consulting the Laplace transform Table 2.1

(entry number 22). Figure 4.9 shows such a response for the prototype system with $\omega_n = 1$ and δ set to 0.2, 0.5, 1 and 1.5, which is obtained by using the MATLAB Script given at the end of the chapter.

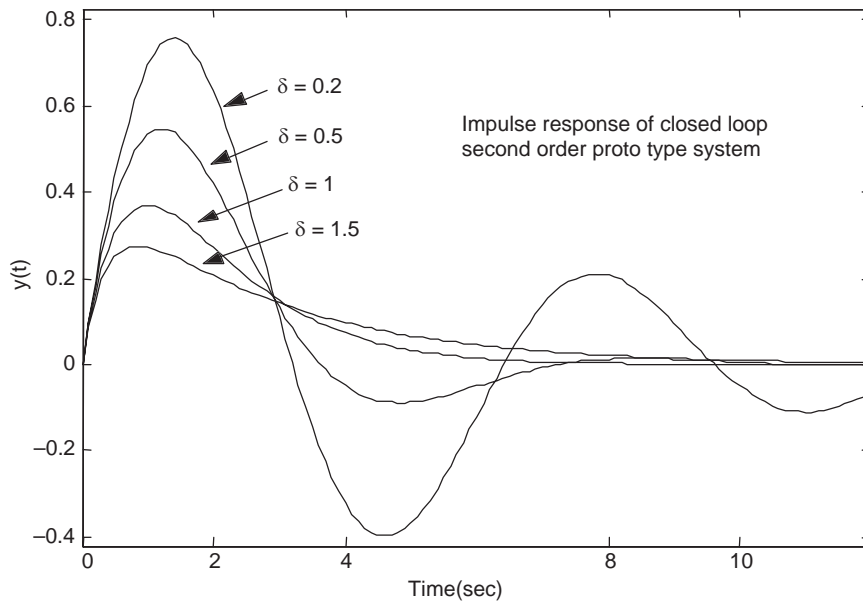



Fig. 4.9 Impulse response of closed loop second order prototype

4.7 THE STEADY-STATE ERROR

Steady-state error in a control system is a measure of system performance when the transient phase of response is over. It indicates the accuracy with which the system output can track a given input. It is desirable that the steady state error should be zero. However, in a real system, the output in the steady state seldom reaches the set value due to friction in a mechanical system and other limitations in the system characteristics. So, the steady state error is unavoidable in most control systems and it is one of the important objectives of controller design to keep this error below a certain acceptable limit, if it can not be completely eliminated.

In Section 1.5 of Chapter 1, we had a glimpse of the areas where feedback control systems are employed, and it gives an idea about the divergent nature and types of physical variables that are controlled automatically. The steady state error is associated with each and every physical entity to be controlled .There may be the steady state error in the control of temperature, pressure, flow, frequency, position, velocity, voltage, current, torque and so on.

The tolerable limits in steady state error in a control system depend on the particular application. Take, for instance, an elevator control system where the floor of the elevator should be aligned level with the floor of a building. Because of a dead zone in the controller (d in Fig. 4.10), there will be some error in alignment between the level of the floor of the building and the floor of the elevator platform when the elevator comes to rest following the command from the operator. If the elevator is intended to transport normal passengers only, an error not exceeding 3 inches may possibly be tolerated, where the passengers can negotiate the 3 inch 'step' between the building floor and elevator floor. However, if it is to be used by people in a wheel chair, an error more than a fraction of an inch is not acceptable. Even when the elevator is used to ferry cars in a multi-storied parking center, a steady state error of 3 inches may be a problem for cars with small wheel-diameters. An error of one thousandth of a radian can not

be tolerated in antennae position control systems mounted on satellites for critical astronomical studies.

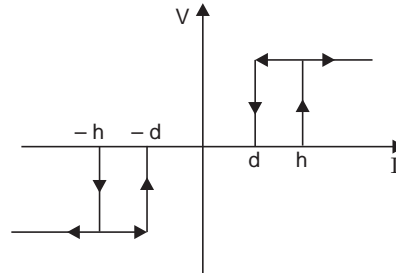



Fig. 4.10 Characteristic of a electro-mechanical relay with dead zone and hysteresis

4.7.1 Steady-State Error Caused by Nonlinear Elements

We mentioned that the system characteristics like dead zone and friction is responsible for steady-state errors in control systems. In the elevator control system example, if the actuator for applying power to the hauling motor has a characteristic as shown in Fig. 4.10, then whenever the error amplifier current I is within the dead zone $\pm d$, the actuator output is zero and the motor torque developed is also zero. So, when the error current enters the dead zone area during the transient state, the elevator carriage will freely coast to a position, determined by the inertia of the carriage and the friction associated with the moving mass and the control would not be able to correct any error present.

In control of mechanical objects, the coulomb friction is, very often, responsible for steady-state errors in position servo systems.

Digital control systems employs microcomputers, where the signals can take on only discrete or quantized values. The analog signal is converted to discrete signals by analog to digital converter (ADC) and the output of the ADC will be zero when the amplitude of the analog input is less than the least significant number that can be represented by the finite bit ADC output. This will give rise to steady state error in the discrete closed loop system.

4.8 STEADY-STATE ERROR OF LINEAR CONTROL SYSTEMS

An important performance criteria for a control system is to minimize the system error in the steady state for typical test inputs. In general, the steady-state errors of linear control systems depend on the types of input and the system characteristic. If the reference input $r(t)$ and the controlled output $y(t)$ in a system with unity feedback, are of the same physical nature or dimension, say, position, frequency, voltage or any other derived units, the error is simply defined as

$$e(t) = r(t) - y(t) \tag{4.20}$$

Now consider the case of a speed control system of a DC motor, where for ease of signal processing and to avoid other difficulties, a reference angular velocity is represented by a reference voltage r and the angular position of the motor is represented by $y(t)$. In this case, the error signal cannot be defined simply as the difference between $r(t)$ and $y(t)$ as in Equation (4.20), because they represent physical entities with different dimensions. A non-unity feedback element, H(s), is to be introduced in the system as shown in Fig. 4.4, for converting $y(t)$ to voltage proportional to speed, so that the dimension of $r(t)$ and $b(t)$ is the same and they can be compared such that the error can be defined as

$$e(t) = r(t) - b(t)$$

$$= r(t) - K_t \frac{dy(t)}{dt} \quad (4.21)$$

The feedback element in this case is a tachogenerator with output voltage proportional to angular speed, *i.e.,* $b(t) = K_t \frac{dy(t)}{dt}$, so that the transfer function of the feedback element, with zero initial value, is given by $H(s) = K_t s$

and $$E(s) = R(s) - B(s) = R(s) - H(s)Y(s) \quad (4.22)$$

The error in (4.22) becomes zero when the output velocity $dy(t)/dt$ is equal to $r(t)/K_t$.

The steady-state error of a feedback control system can, therefore, be defined as

$$\text{Steady-state error} = e_{ss} = \lim_{t\to\infty} e(t) \quad (4.23)$$

With reference to Fig. 4.4, the Laplace transform of the error signal is

$$E(s) = \frac{R(s)}{1 + G(s)H(s)} \quad (4.24)$$

By use of the final-value theorem of the Laplace transform (vide Section 2.4 in Chapter 2), the steady-state error of the system is

$$e_{ss} = \lim_{t\to\infty} e(t) = \lim_{s\to 0} sE(s) \quad (4.25)$$

where it is assumed that $sE(s)$ converges, that is, $sE(s)$ does not have any pole on the imaginary axis or on the right-half of the s-plane. Substituting the value of $sE(s)$ from Equation (4.24) into Equation (4.25), we get

$$e_{ss} = \lim_{s\to 0} \frac{sR(s)}{1 + G(s)H(s)} \quad (4.26)$$

Equation (4.26) shows the dependence of steady-state error on the reference input $R(s)$ and the loop transfer function $G(s)H(s)$.

It is to be noted here, that even though the error function is defined with reference to the system configuration shown in Fig. 4.4, the term error may also be used to describe signals in other parts of the system. When an unwanted disturbance acts on the plant, then the plant response due to this perturbation alone is also referred to as error.

4.8.1 The Type of Control Systems

When the control system is represented by the simplified block diagram of Fig. 4.4, the steady-state error is given by Equation (4.26), which is found to depend on the type of input and the loop gain $G(s)H(s)$. Let us represent the general form of the loop gain $G(s)H(s)$ as

$$G(s)H(s) = \frac{K(s + z_1)(s + z_2)\ldots(s + z_m)e^{-T_d s}}{s^q(s + p_1)(s + p_2)\ldots(s + p_n)} : (n > m) \quad (4.27)$$

where $p_1, p_2, \ldots p_n$ and $z_1, z_2 \ldots.., z_m$ are distinct poles and zeros of $G(s)H(s)$ and the pole at the origin $s = 0$ is repeated q times. Now, the *type* of the closed-loop system refers to q , the *order* of the pole of $G(s)H(s)$ at the origin. So, the closed-loop system with loop transfer function in Equation (4.27) is of type q, where $q = 0,1,2,\ldots.$ The *type* of a system does not depend on K or the location and count of other poles and zeros but solely on q, the order of the pole at the

origin of the loop gain. The following example illustrates the system types with reference to the form of G(s)H(s).

$$\text{Type 1 system} \qquad G(s)H(s) = \frac{K(s+2)}{s\,(s+1)\,(s+5)} \tag{4.28}$$

$$\text{Type 2 system} \qquad G(s)H(s) = \frac{K(s+1)}{s^2(s+2)} \tag{4.29}$$

Since the steady state error is found to depend on the input as well as on the system types, we shall now investigate the effects of some typical inputs like step, ramp and parabolic, on the steady-state errors.

The definitions of a few terms related to steady state error are in order.

4.8.2 Steady-State Error of a System with a Step-Function Input

Let us consider that the reference input to the system of Fig. 4.4 is a step function with amplitude R, so that the Laplace transform R(s) is given by R/s. Using this value of R(s) in the Equation (4.26), the steady state error given by

$$e_{ss} = \lim_{s\to 0} \frac{s\,R(s)}{1+G(s)\,H(s)} = \lim_{s\to 0} \frac{R}{1+G(s)\,H(s)} \tag{4.30}$$

or

$$e_{ss} = \frac{R}{1+\lim\limits_{s\to 0} G(s)\,H(s)} = \frac{R}{1+K_p} \tag{4.31}$$

where,

$$K_p = \lim_{s\to 0} G(s)\,H(s) \tag{4.32}$$

The constant K_p is referred to as the *step-error (or position error) constant.*

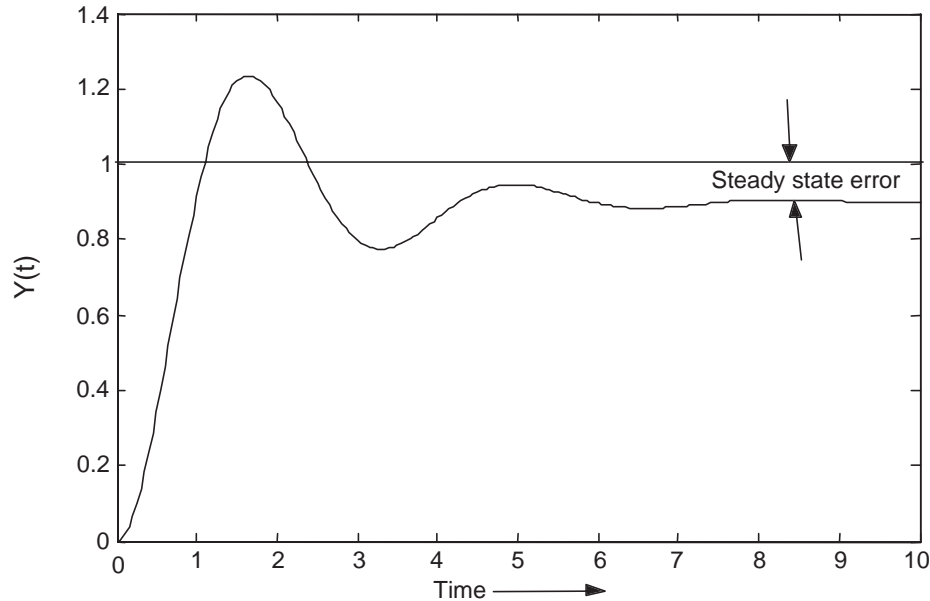



Fig. 4.11 Steady state error of type 0 system when subjected to a step input

From Equation (4.31) it is evident that the steady state error e_{ss} will be zero if K_p is infinite and K_p will be infinite if q in relation (4.27) is at least 1. So, for the steady state error to be zero with a step input, the system type should be at least 1 or higher. That is, for a step input the steady state error is related to system type as:

Type 0 system: $$e_{ss} = \frac{R}{1+K_p} \text{ finite constant} \tag{4.33}$$

Type 1 or higher system: $e_{ss} = 0$

Fig. 4.11 shows the finite steady state error of a typical system subjected to a step input.

The $1/s$ term in the forward path represents the transfer function of an integrator, so any finite error, however small, is integrated over time to force the actuator to act on the process to move the output towards the set value making the error to be zero in the steady state.

4.8.3 Steady-State Error of A System with Ramp-Function Input

If the input to the control system of Fig. 4.4 is a ramp function given by:

$$r(t) = Rtu_s(t) \tag{4.34}$$

where R is a real constant, the Laplace transform of $r(t)$ is found to be

$$R(s) = \frac{R}{s^2} \tag{4.35}$$

Substituting the value of R(s) from Equation (4.35) into Equation (4.26), we get the steady state error as:

$$e_{ss} = \lim_{s\to 0} \frac{R}{s + sG(s)H(s)} = \frac{R}{\lim_{s\to 0} sG(s)H(s)} = \frac{R}{K_v} \tag{4.36}$$

where $$K_v = \lim_{s\to 0} s\,G(s)H(s) \tag{4.37}$$

The parameter K_v in relation (4.37) is known as the **ramp-error (or velocity error) constant.**

The relation (4.36) shows that for the steady state error e_{ss} to be zero for the ramp input function, K_v must be infinite. Using Equations (4.37) and (4.27), we get

$$K_v = \lim_{s\to 0} s\,G(s)H(s) = \lim_{s\to 0} \frac{K}{s^{j-1}}\; j = 0, 1, 2, \ldots \tag{4.38}$$

Thus we see that for K_v to be infinite, j must be at least equal to 2 or higher. So, we have the following conclusions about the steady-state error of a system with a ramp input:

System *type* 0: $e_{ss} = \infty$

System *type* 1: $$e_{ss} = \frac{R}{K_v} \tag{4.39}$$

System *type* 2 *or higher* 0: $e_{ss} = 0$

A typical steady state error e_{ss} for a ramp input with finite K_v is shown in Fig. 4.12. It is to be noted that K_v gives a measure of the steady-state error in position for velocity (ramp) type of input and has nothing to do with error in velocity.

Chapter 4

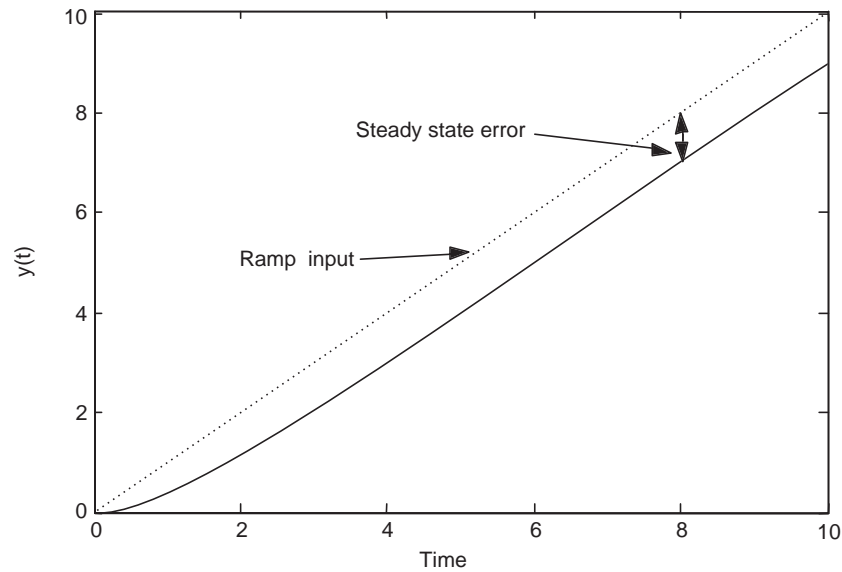



Fig. 4.12 Steady state error of type 1 system when subjected to a ramp input

4.8.4 Steady-State Error of A System with Parabolic-Function Input

If parabolic input is described by

$$r(t) = \frac{Rt^2}{2}\, u_s(t) \tag{4.40}$$

The Laplace transform of $r(t)$ will be given by

$$R(s) = \frac{R}{s^3} \tag{4.41}$$

Using this value of R(s) in relation (4.26), the steady state error for the system in Fig. 4.6 is found to be

$$e_{ss} = \frac{R}{\lim\limits_{s \to 0} s^2 G(s)H(s)}$$

or
$$e_{ss} = \frac{R}{K_a} \tag{4.42}$$

where
$$K_a = \lim_{s \to o} s^2 G(s)H(s) \tag{4.43}$$

The parameter K_a is referred to as the **parabolic-error constant** and is a measure of error in 'position' due to 'acceleration' input.

So, we can conclude about the steady state error of a system in Fig. 4.4 with parabolic input as follows:

System *type* 0: $e_{ss} = \infty$

System *type* 1: $e_{ss} = \infty$

System *type* 2: $e_{ss} = \frac{R}{K_a}$ = constant (4.44)

System *type* 3 or higher: $e_{ss} = 0$

The steady state error e_{ss} of a typical system with parabolic input and finite K_a is shown in Fig. 4.13.

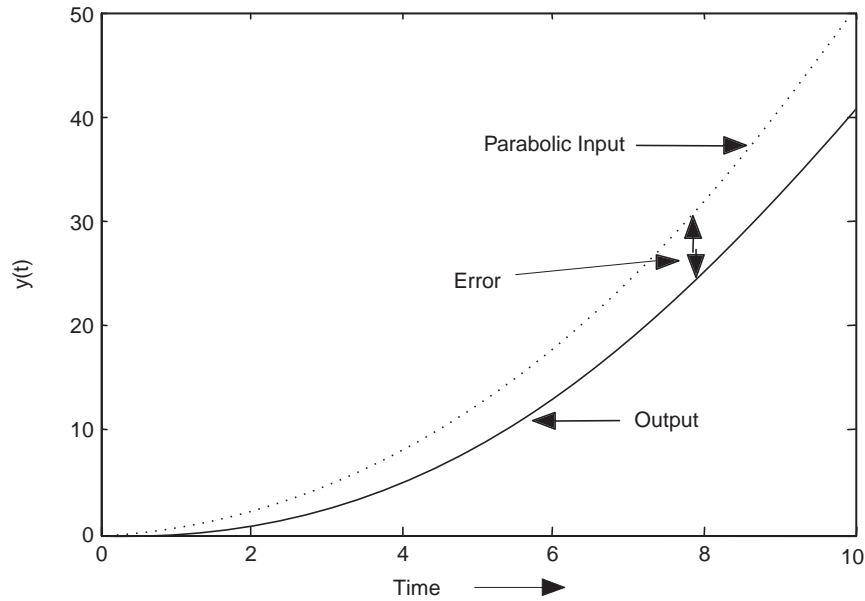



Fig. 4.13 Steady state error of type 2 system when subjected to a parabolic input

4.9 PERFORMANCE INDEXES

We mentioned in Chapter 1 (Section 1.7) that one of the important tasks of a control engineer is to solve a mathematical problem such that a system satisfies a performance specification expressed in mathematical terms. We have already found the conditions for meeting the performance specification on steady state error of a closed loop system subjected to some typical test inputs. We shall now consider some mathematical functions to satisfy the performance of a closed loop system with respect to its error in the transient state. In this context, four most commonly used mathematical functions used as a performance index associated with error of a closed loop system are shown below. The objective is to design an optimal system by proper choice of its parameters such that the specified performance index is extremum-either minimum or maximum depending on the context. The problem of optimal control, on the other hand, does not change the plant-parameters at all, instead, the control signal is designed in such a way as to optimize the system performance with respect to a performance index (vide Chapter 11).

The commonly used performance Indexes (PI) are:

(*a*) Integral of squared error (ISE),

$$J = \int_0^\infty e^2(t)\, dt \tag{4.45}$$

(*b*) Integral of time multiplied squared error (ITSE),

$$J = \int_0^\infty te^2(t)\, dt \tag{4.46}$$

(*c*) Integral of absolute error (IAE),

$$J = \int_0^{\infty} |e(t)|\, dt \quad \text{and} \tag{4.47}$$

(*d*) Integral of time multiplied absolute error(ITAE),

$$J = \int_0^{\infty} t|e(t)|\, dt \tag{4.48}$$

With reference to the system in Fig. 4.4, with H(*s*) = 1, the error $e(t)$ between the output $y(t)$ and the input $r(t)$ is defined as:

$$e(t) = r(t) - y(t) \tag{4.49}$$

If the system is stable and the steady state error is zero *i.e.* $\lim_{t \to \infty} e(t) = 0$ (4.50)

then the performance Indexes (Equations 4.45-4.48) will be finite as t approaches infinity. However, if the steady state error is finite the values of the performance indexes will approach ∞ as t approaches to ∞. In order to keep the values of the performance indexes finite in such cases, we can define error as :

$$e(t) = y(\infty) - y(t) \tag{4.51}$$

There has been extensive studies [75] for designing a class of optimal systems with respect to the foregoing performance indexes. The class of linear systems considered are of the form :

$$\frac{Y(s)}{R(s)} = \frac{a_n}{s^n + a_1 s^{n-1} + \ldots + a_{n-1}s + a_n} \tag{4.52}$$

4.9.1 Integral of Squared Error (ISE)

The performance index ISE in relation (4.45) gives more importance to large errors and less importance to small errors. Besides, this criterion is not very selective with respect to the variation of system parameter. In particular, when this criteria is applied to the second order system given below, with only one design parameter δ with normalized $\omega_n = 1$, a flat range of $\delta = 0.5$ to 0.7 is found that gives more or less the same value of performance index.

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} = \frac{1}{s^2 + 2\delta + 1} \tag{4.53}$$

The step response of the systems designed by minimizing this performance index, is fast but oscillatory in nature, but steady state error is zero. For practical computation of performance index, the upper limit of integration is taken as finite time T, selected in such a way that makes $e(t)$ negligibly small for $t > T$.

4.9.2 Integral of Time Multiplied Squared Error (ITSE) Criteria

The (ITSE) criteria has a characteristic that in the step response of the system a large initial error is weighed lightly because of small value of t, while errors occurring late in the transient response are penalized heavily. This criterion has a better selectivity than the integral square-error criterion.

4.9.3 Integral of Absolute Error (IAE) Criteria

Integral of absolute error criteria is amenable to easy computer calculation but difficult to evaluate analytically. The resulting optimal system that minimizes IAE has reasonable

damping and an acceptable transient-response behavior. It also results in an optimal system that minimizes fuel consumption in systems like space vehicles.

4.9.4 Integral of Time Multiplied Absolute Error (ITAE)

Like the (ITSE) criteria in (*b*), the ITAE criteria also penalizes a large initial error lightly whereas the errors occurring late in the transient response are penalized heavily. An optimal system designed based on ITAE criteria exhibits small overshoot to step response with well damped oscillations. The ITAE criterion also has good selectivity and is, therefore, an improvement over the integral absolute-error criterion. However, it is very difficult to find an analytical expression for a given system , but it is amenable to easy experimental measurements.

Optimal systems having the structure shown in Equation (4.52) designed by the ITAE criteria are shown in the Table 4.1 for $n = 1$ to 6. The step response of these systems are shown in Fig. 4.14. It is to be noted that the steady state error in response to a step input is zero for this class of optimal system.

Table 4.1 Closed-loop transfer function of a class of optimal system that minimizes ITAE criterion

$$\frac{Y(s)}{R(s)} = \frac{a_n}{s^n + a_1 s^{n-1} + \ldots + a_{n-1}s + a_n}, \; a_n = \omega_n^{\,n}$$

Order of the system, n	*Denominator of optimal transfer function*
1	$s + \omega_n$
2	$s^2 + 1.4\omega_n s + \omega_n^2$
3	$s^3 + 1.75\omega_n s^2 + 2.15\omega_n^2 s + \omega_n^3$
4	$s^4 + 2.1\omega_n s^3 + 3.4\omega_n^2 s^2 + 2.7\omega_n^3 s + \omega_n^4$
5	$s^5 + 2.8\omega_n s^4 + 5.0\omega_n^2 s^3 + 5.5\omega_n^3 s^2 + 3.4\omega_n^4 s + \omega_n^5$
6	$s^6 + 3.25\omega_n s^5 + 6.60\omega_n^2 s^4 + 8.60\omega_n^3 s^3 + 7.45\omega_n^4 s^2 + 3.95\omega_n^5 s + \omega_n^6$

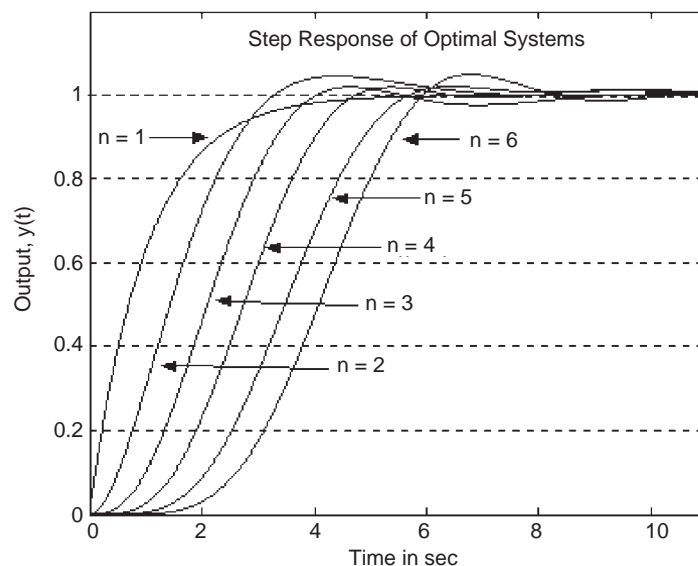



Fig. 4.14 Step response of optimal closed loop system designed to minimize ITAE criterion

4.9.5 Quadratic Performance Index

The performance index most widely used for designing linear optimal control system is of the form:

$$J = \frac{1}{2}\left\{\int_{t_i}^{t_f} [x'(t)Q(t)\,x(t) + u'(t)\,R(t)\,u(t)]\,dt + x'(t_f)\,S_f\,(t_f)\right\} \tag{4.54}$$

where $x(t)$ is the n-dimensional state vector (which may include error vector), $Q(t) = Q' \geq 0$ is the positive semi-definite matrix associated with the weights of the states and the positive definite matrix $R(t) = R'(t) > 0$ is the weights given to the control vector. The symmetric matrix $S_f = S'_f \geq 0$ is the weight assigned to the terminal state. The term $u'(t)R(t)u(t)$ in the above performance index is, therefore, a measure of control energy. The limits of the integration are the initial time t_i and the terminal time t_f which may be infinite for the regulatory system where the reference input is either constant or varying very slowly.

The problem of interest is the optimal control of the linear system described by

$$\dot{\mathbf{x}}(t) = A(t)\mathbf{x}(t) + B(t)\mathbf{u}(t) \tag{4.55}$$

$$\mathbf{x}(t_i) = \mathbf{x}_o$$

with given initial state x_o *and time* t_i *a control is to be found which minimizes the performance index.*

The final time t_f *is given, while no constraints are imposed on the final state* $x(t_f)$*. The matrices* A, B, Q *and* R *in Equations (4.54) and (4.55) are continuously differentiable functions.* [here $Q'(t)$ stands for the transpose of the matrix $Q(t)$].

The solution of optimal control problem is taken up in Chapter 11.

4.10 FREQUENCY DOMAIN RESPONSE

The performance of a control system is normally measured in terms of its time-domain characteristics. It may equally be measured in frequency-domain, since there are correlations between the frequency-domain and time-domain performances in a linear system, so that the time-domain properties of the system can be predicted based on the frequency-domain analysis. Besides, the frequency domain approach provides an alternative perspective to the complex analysis and design problems of linear control systems in addition to the time-domain studies. The frequency-domain analysis of a linear control system assumes that the input is sinusoidal, even though, it does not necessarily imply that the system will be subjected to a sine-wave input only. Rather, the engineer will be able to project the time-domain performance of the system from the frequency-domain studies.

Assume that a linear time-invariant system with transfer function $G(s)$ is subjected to sinusoidal input with amplitude R and frequency ω_0 given by

$$r(t) = R \sin \omega_o t \tag{4.56}$$

The steady-state output of the system $y(t)$ will also be a sinusoid with the same frequency ω_o, but possibly with different amplitude and phase; *i.e.*

$$y(t) = Y \sin (\omega_o t + \phi) \tag{4.57}$$

where Y is the amplitude of the output sine wave and ϕ is the phase shift in degrees or radians introduced by the system.

The Laplace transform of the output of a linear single-input single-output system with transfer function G(s) can be expressed in terms of the input from the definition of transfer function(vide Section 2.6) as:

$$Y(s) = G(s)R(s) \tag{4.58}$$

We know, in general that $s = \sigma + j\omega$ (vide Section 2.0 in Chapter 2). However, for sinusoidal steady-state analysis, we shall replace s by its imaginary component $j\omega$ only, since in the steady state, the contribution of the real part σ will disappear for a stable system. So with $s = j\omega$, the Equation (4.58) becomes

$$Y(j\omega) = G(j\omega)R(j\omega) \tag{4.59}$$

By writing the functions G($j\omega$) and R($j\omega$) in terms of their amplitudes and phases, respectively, as

$$Y(j\omega) = | G(j\omega) | \angle G(j\omega) \tag{4.60}$$

and

$$R(j\omega) = | R(j\omega) | \angle R(j\omega) \tag{4.61}$$

the relation (4.60) can be expressed in terms of overall amplitude relation between input and output of the linear system (4.59)

$$| Y(j\omega) | = | G(j\omega) | | R(j\omega) | \tag{4.62}$$

and the phase relation

$$\angle Y(j\omega) = \angle G(j\omega) + \angle R(j\omega) \tag{4.63}$$

Thus, for the input and output signals described by Equations (4.56) and (4.57) respectively, the amplitude of the output sinusoid is

$$Y = R \, | G(j\omega_o) | \tag{4.64}$$

and the phase of the output is

$$\phi = \angle G(j\omega_o) \tag{4.65}$$

Thus, by knowing the transfer function G(s) of a linear system, the magnitude characteristics, $|G(j\omega)|$, and the phase characteristics, $\angle G(j\omega)$, completely describe the steady-state performance for sinusoidal input. The important feature of the frequency domain analysis of linear systems is that the amplitude and phase characteristics of a closed-loop system can be used to predict the time-domain transient performance characteristics such as overshoot, rise time and the steady-state performance.

4.10.1 Frequency Response of Closed-Loop Systems

The closed loop transfer function of system in Fig. 4.6 is given by:

$$M(s) = \frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \tag{4.66}$$

Under the sinusoidal steady state, $s = j\omega$; Equation (4.66) becomes

$$M(j\omega) = \frac{Y(j\omega)}{R(j\omega)} = \frac{G(j\omega)}{1 + G(j\omega)H(j\omega)} \tag{4.67}$$

In terms of its magnitude and phase, the sinusoidal steady-state transfer function M($j\omega$) may be expressed as:

$$M(j\omega) = | M(j\omega) | \angle M(j\omega) \tag{4.68}$$

Or in terms of its real and imaginary parts of M($j\omega$) as:

$$M(j\omega) = \text{Re}\,[M(j\omega)] + j\text{I}m[M(j\omega)] \tag{4.69}$$

The magnitude of M($j\omega$) is

$$| M(j\omega) | = \left| \frac{G(j\omega)}{1 + G(j\omega)H(j\omega)} \right| = \frac{| G(j\omega) |}{| 1 + G(j\omega)H(j\omega) |} \tag{4.70}$$

and the phase of $M(j\omega)$ is

$$\angle M(j\omega) = \phi_M(j\omega) = \angle G(j\omega) - \angle [1 + G(j\omega)H(j\omega)] \tag{4.71}$$

It is clear from Equations (4.70) and (4.71) that the gain and phase characteristics of the closed-loop system can be determined from that of the open-loop transfer function $G(s)$ as well as the loop transfer function $G(s)H(s)$. In practice, the frequency responses of $G(s)$ and $H(s)$ can be determined experimentally by applying sine-wave inputs to the system and sweeping the frequency from 0 to a value beyond the frequency range of the system.

4.10.2 Frequency-Domain Specifications

In the design of linear control systems using the frequency-domain methods, it is necessary to define a set of specifications so that the performance of the system can be properly assessed in terms of these specifications. The following frequency-domain specifications are often used in practice.

(*a*) **Resonance Peak M_r:** *This is the maximum value of the amplitude M and it indicates a measure of relative stability*.

The large value of M_r is associated with large overshoot in transient input response. Recommended value of M_r is in the range of 1.1 to 1.5 for a closed loop system.

(*b*) **Resonant Frequency ω_r:** *The resonant frequency ω_r is defined as the frequency at which the peak amplitude M_r* occurs.

(*c*) **Bandwidth:** The bandwidth (BW) is *defined as the frequency at which the magnitude of $|M(j\omega)|$ drops to 70.7 percent of its zero-frequency value, or 3 dB down from the zero-frequency value.* In general, the bandwidth of a control system gives a measure of the transient-response properties, in that a large bandwidth corresponds to a faster rise time, since higher-frequency signals are more easily passed on to the outputs. Conversely, if the bandwidth is small, signals of relatively low frequencies only can pass to the output which makes the time response sluggish. Bandwidth also indicates the noise-filtering characteristics and the robustness of the system.

(*d*) **Cutoff Rate:** Often, bandwidth alone is inadequate in the indication of the characteristics of the system in distinguishing signals from noise. Sometimes it may be necessary to specify the cutoff rate of the frequency response, which is the slope of $|M(j\omega)|$ at the high frequency end of its spectrum. Apparently, two systems can have the same bandwidth, but the cutoff rates of the frequency responses may be different.

The performance measures for the frequency-domain analysis are illustrated in Fig. 4.15. There are other criteria that are just as important in specifying the relative stability and performance of a control system in the frequency domain. These are defined in Section 6.3 of Chapter 6.

4.11 FREQUENCY DOMAIN PARAMETERS OF PROTOTYPE SECOND-ORDER SYSTEM

4.11.1 Peak Resonance and Resonant Frequency

Some of the parameters of the frequency-domain specifications can be expressed in a closed form for a prototype second order system. Let us consider the closed-loop transfer function of a prototype second order system given by:

$$\frac{Y(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\delta\omega_n s + \omega_n^2} \tag{4.72}$$

We get the steady state frequency-domain value from the transfer function by substituting $s = j\omega$ in M(s) as shown below:

$$M(j\omega) = \frac{Y(j\omega)}{R(j\omega)} = \frac{\omega_n^2}{(j\omega)^2 + 2\delta\omega_n(j\omega) + \omega_n^2}$$

$$= \frac{1}{1 + j2(\omega/\omega_n)\delta - (\omega/\omega_n)^2} \tag{4.73}$$

We can simplify the Equation (4.73) by letting $u = \omega/\omega_n$, to get

$$M(ju) = \frac{1}{1 + j2u\delta - u^2} \tag{4.74}$$

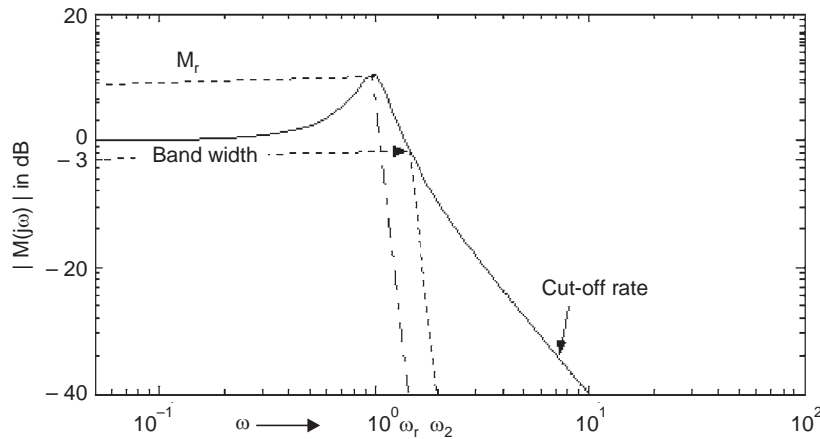



Fig. 4.15 Frequency-domain specifications

The magnitude and phase of M(ju) are

$$|M(ju)| = \frac{1}{[(1-u^2)^2 + (2\delta u)^2]^{1/2}} \tag{4.75}$$

and

$$\angle M(ju) = \phi_m(u) = -\tan^{-1}\frac{2\delta u}{1-u^2} \tag{4.76}$$

respectively. The resonant frequency is determined by setting the derivative of | M(ω)| with respect to u to zero.

This yields $$u = \sqrt{1-2\delta^2} \tag{4.77}$$

and the corresponding resonant frequency is given by $\omega_r = \omega_n\sqrt{1-2\delta^2}$ (4.78)

Since frequency is a real quantity, Equation (4.77) is valid only for $1 \geq 2\delta^2 > 0$ or $0 < \delta \leq 0.707$. Therefore, resonance will not occur for any value of δ greater than 0.707 and the magnitude M of the frequency response is less than unity for $\delta > 0.707$.

Substituting Equation (4.77) in Equation (4.75) for u and simplifying, we get

$$M_r = \frac{1}{2\delta\sqrt{1-\delta^2}}, \quad 0 < \delta \leq 0.707 \tag{4.79}$$

So for the prototype second-order system described by Equation (4.72), the resonance peak M_r is a function of the damping ratio δ only, and ω_r is a function of δ and ω_n. Furthermore, although taking the derivative of $|M(ju)|$ with respect to u is a valid method of determining M_r and ω_r, in general, for high-order systems, this analytical method is quite tedious and is not recommended. The graphical methods that will be discussed in this chapter and the Control toolbox of MATLAB is very convenient for the frequency-domain analysis and design of linear control systems.

Figure 4.16(*a*) illustrates the plots of $|M(ju)|$ of Equation (4.75) with normalized frequency u for various values of δ. Figure 4.16(*b*) shows the plot $u = \frac{\omega_r}{\omega_n}$ with δ. Also We note that when δ = 0, the resonance frequency ω_r becomes equal to ω_n.

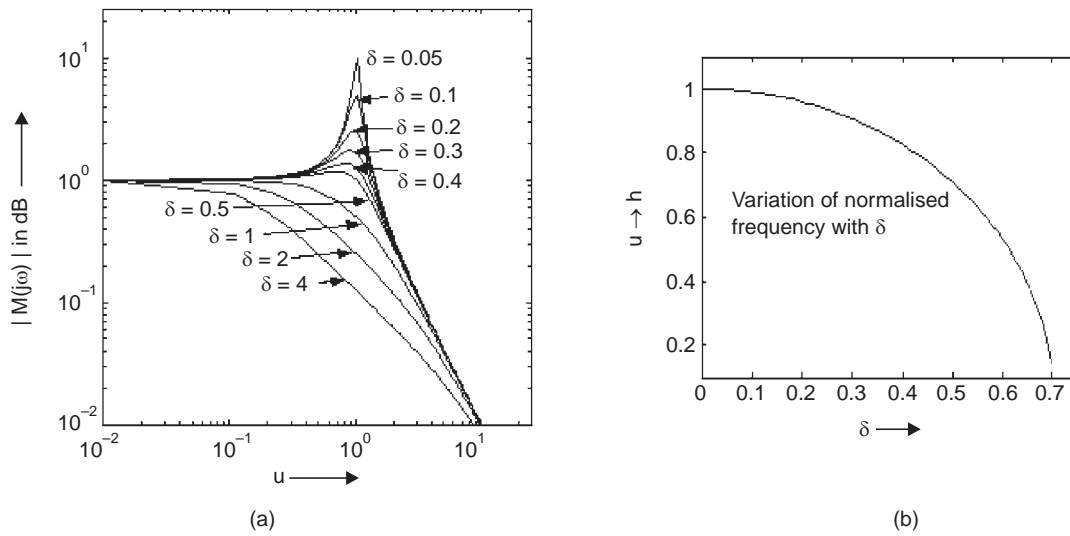



Fig. 4.16 (a) Plots of | M(*ju*) | with *u* for various values of damping ratios
(b) Plot of normalized frequency u with damping ratio

4.11.2 Bandwidth

The bandwidth BW of a closed-loop system is the frequency at which $|M(j\omega)|$ drops to 70.7 percent of, or 3 dB down, from the zero-frequency value. Equating Equation (4.75) to 0.707, we have

$$|M(ju)| = \frac{1}{[(1-u^2)^2 + (2\delta u)^2]^{1/2}} = 0.707 \tag{4.80}$$

Thus,
$$[(1-u^2)^2 + (2\delta u)^2]^{1/2} = 1.414 \tag{4.81}$$

Squaring both sides and rearranging terms, we get

$$u^2 = (1-2\delta^2) \pm \sqrt{4\delta^4 - 4\delta^2 + 2} \tag{4.82}$$

The positive sign in the right hand side in the above expression will give the upper half power frequency u_2 whereas the negative sign will give lower half power frequency u_1. But the lower half power frequency is computed as negative for any δ > 0, which is not acceptable. The lower half power frequency u_1 is taken as 0.

So, the bandwidth $\omega_2 - \omega_1 = \omega_n(u_2 - u_1)$ of the prototype second-order system is found as

$$BW = \omega_2 - \omega_1 = \omega_n[(1 - 2\delta^2) + \sqrt{4\delta^4 - 4\delta^2 + 2}\,]^{1/2} \quad (4.83)$$

We see from Equation (4.83) that for a fixed ω_n, the bandwidth increases as the damping ratio δ decreases from unity. Fig. 4.17 shows a plot of bandwidth with damping ratio.

So, we have established some simple relationships between the time-domain response and the frequency-response characteristics of the prototype second-order system that will be helpful in the design of linear control systems.

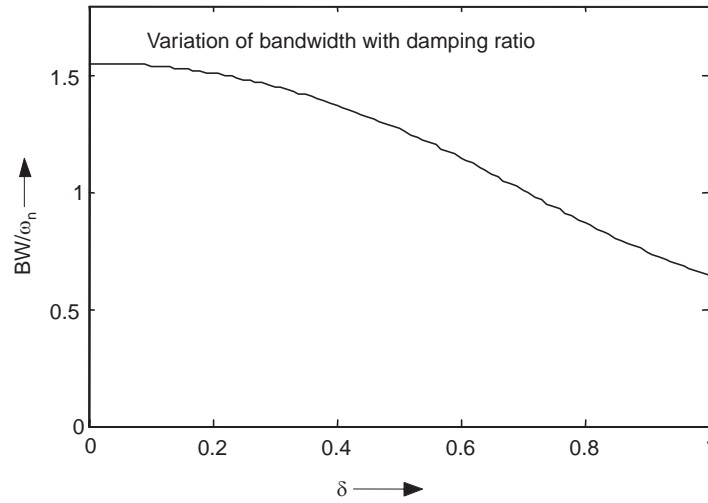



Fig. 4.17 Plot of bandwidth with damping ratio for prototype second order system

4.12 BODE DIAGRAMS

In Section 4.6 we have noted that the complex function $G(j\omega)H(j\omega)$ is characterized by its magnitude and phase angle, with frequency as the parameter. There are three commonly used representations of $G(j\omega)H(j\omega)$. These are:

1. Bode diagram or logarithmic plot
2. Polar plot
3. Log-magnitude versus phase plot

This section presents Bode diagrams of $G(j\omega)H(j\omega)$. Polar plots and log-magnitude versus phase plots are presented in Chapter 6.

4.12.1 Bode Plot

The complex function $G(j\omega)H(j\omega)$ may be represented by two separate graphs, one giving the magnitude versus frequency and the other the phase angle (in degrees) versus frequency. For convenience, the logarithm of the magnitude $G(j\omega)H(j\omega)$ is employed in Bode diagram and the frequency is also plotted in logarithmic scale, both for the amplitude as well as the phase angle.

The standard representation of the logarithmic magnitude of $G(j\omega)H(j\omega)$ is $20 \log |G(j\omega)H(j\omega)|$, where the base of the logarithm is 10. The unit used in this representation of the magnitude is the decibel, abbreviated as dB. In the logarithmic representation, the curves are drawn on semi-log paper, using the log scale for frequency and the linear scale for either magnitude (but in db) or phase angle (in degrees).

A very attractive advantage of using logarithmic plot for the frequency response of transfer function is that multiplication of magnitudes contributed by individual factors can be converted into addition. Furthermore, it permits a simple method for obtaining an approximate log-magnitude curve of the transfer function. It is based on straight-line asymptotic approximations, which is adequate for the rough sketch of the frequency response characteristics needed in the design stage. And when exact curves are needed, corrections can be easily incorporated to these basic asymptotic plots.

The use of logarithmic scale makes it possible to display both the low- and high-frequency characteristics of the transfer function in one graph. Even though zero frequency cannot be included in the logarithmic scale (since log 0 = – ∞), it is not a serious problem as one can go to as low a frequency as is required for analysis and design of practical control system.

Besides, it is convenient to use Bode diagram for plotting the frequency-response data obtained experimentally for a system for the estimation of transfer function.

4.12.2 Principal Factors of Transfer Function

The principal factors that may be present in a transfer function $F(j\omega) = G(j\omega)H(j\omega)$, in general, are:

1. Constant gain K
2. Pure integral and derivative factors $(j\omega)^{\pm n}$
3. First-order factors $(1 + j\omega T)^{\pm 1}$
4. Quadratic factors $[1 + 2\delta(j\omega/\omega_n) + (j\omega/\omega_n)^2]^{\pm 1}$

Once the logarithmic plots of these basic factors are known, it will be convenient to add their contributions graphically to get the composite plot of the multiplying factors of $G(j\omega)H(j\omega)$, since the product of terms become additions of their logarithms.

It will be discovered soon that in the logarithmic scale, the actual amplitude and phase plots of the principal factors of $G(j\omega)H(j\omega)$ may be approximated by straight line asymptotes, which is the added advantage of the Bode plot. The errors in the approximation in most cases are definite and known and when necessary, corrections can be incorporated easily to get an accurate plot.

(A) Real constant gain K:

$$F(j\omega) = K$$

$$F_{db} = 20 \log_{10} K \tag{4.84a}$$

and

$$\angle F = \begin{cases} 0, & : K > 0 \\ -180^\circ & : K < 0 \end{cases} \tag{4.84b}$$

(B) Pure integral and derivative factors (Pole and Zero at the origin):

$$F(j\omega) = (j\omega)^{\pm n}$$

The magnitude

$$F_{db} = \pm n\ 20 \log_{10}\omega \quad \text{for } \omega > 0 \tag{4.85a}$$

and

$$\angle F = \pm n \times 90 \tag{4.85b}$$

Now taking $\log_{10} \omega$ as the x-axis and F_{db} as the y-axis, relation (4.85a) represents a straight line with a slope of $\pm n$ 20dB/decade, this is because of the fact that a unity change of $\log_{10} \omega$ corresponds to change of 10 in ω, that is, a change from 2 to 20 or from 10 to 100. A few amplitude plots for n = 1, 2, – 1, – 2 are shown in top part of Fig. 4.18. The amplitude plots cross the 0 db line at ω = 1. The phase plots are shown in the bottom part of Fig. 4.18.

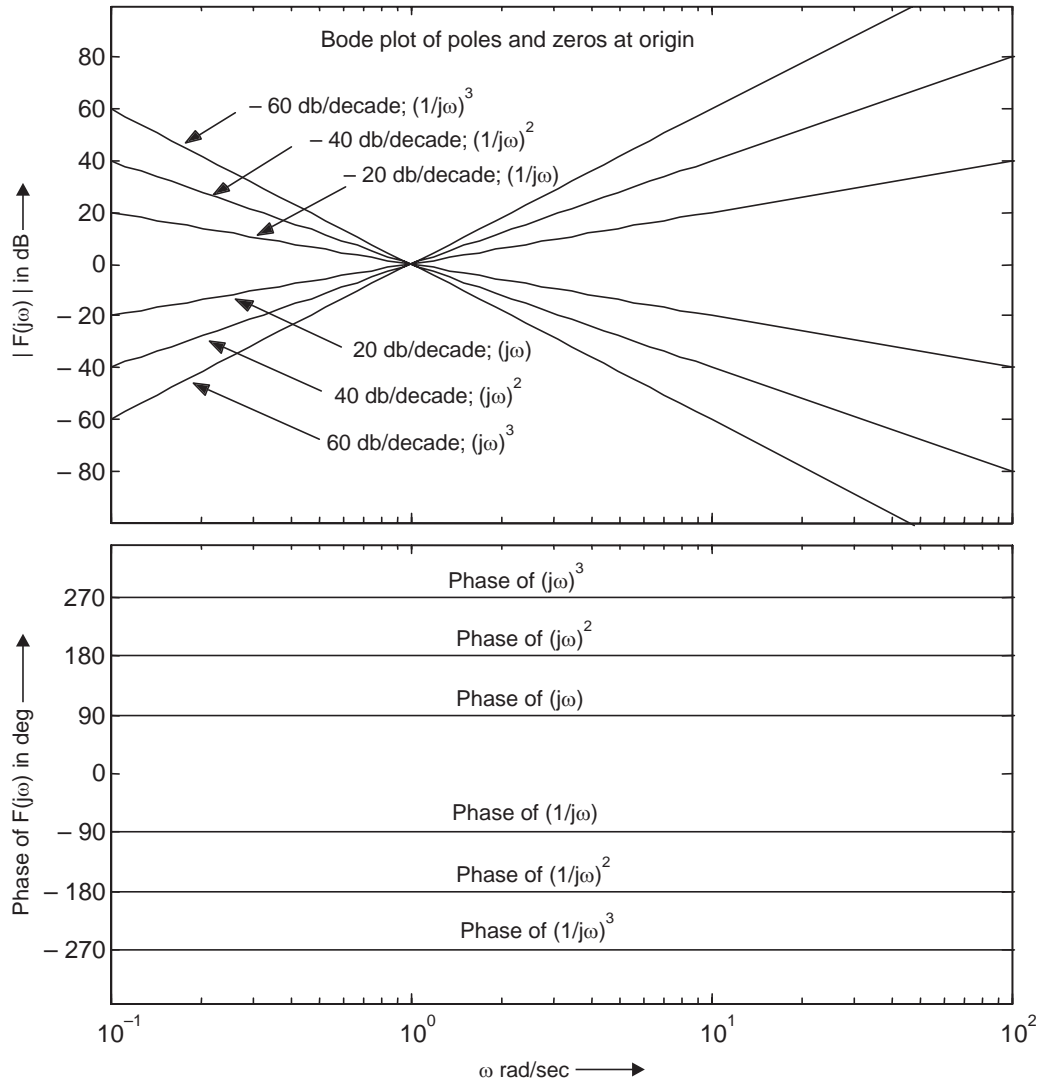



Fig. 4.18 Amplitude and phase plots of poles and zeros at the origin

The slopes are also expressed in octaves. The two frequencies ω_1 and ω_2 are separated by one octave if $\omega_2 / \omega_1 = 2$ and they are separated by a decade if $\omega_2/\omega_1 = 10$. So in $\log_{10} \omega$ scale, the number of decades between any two arbitrary frequencies ω_3 and ω_4 is obtained as

$N_{decades} = \dfrac{\log_{10} \dfrac{\omega_2}{\omega_1}}{\log_{10} 10} = \log_{10} \dfrac{\omega_2}{\omega_1}$, whereas the number of octaves between them is found as

$N_{octaves} = \dfrac{\log_{10} \dfrac{\omega_2}{\omega_1}}{\log_{10} 2} = 3.01 \log_{10} \dfrac{\omega_2}{\omega_1} = 3.01\ N_{decades}$ (using the above relation). So the slope of $\pm\, n$ 20 dB/decade is equivalent to $\pm\, n$ 6 dB/octave.

(C) Factors corresponding to simple poles and zeros: $F(j\omega) = (1 + j\omega T)^{\pm 1}$

The log magnitude of the first-order factor is given by

Amplitude $F_{db} = \pm 20 \log_{10} \sqrt{1 + (\omega T)^2}$

$$= \begin{cases} \pm 20 \log_{10} \omega T, & \omega T >> 1 \\ 0 & , \quad \omega T << 1 \end{cases} \quad (4.86)$$

The amplitude plot may be approximated by two straight lines, (*i*) one with a slope of ± 20 db/decade or ± 6 db/octave and passing through frequency ω = 1/T, known as corner frequency, (*ii*) the other is coincident with 0 db line. The actual plot approaches asymptotically to these straight lines and has a maximum error of ± 3 db occurring at the corner frequency. It is to be noted that the positive sign in the above and subsequent relations are considered for positive power of the factor whereas the negative sign are to be taken for negative power of the factor. The phase angle of first-order factor is given by:

$$\angle F(j\omega) = \pm \tan^{-1}(\omega T)$$

$$= \begin{cases} 90° & , \text{ for } \omega T >> 10 \\ 0° & , \text{ for } \omega T << 0.1 \end{cases} \quad (4.87)$$

The phase angle may also be approximated for $\frac{0.1}{T} < \omega < \frac{10}{T}$ by a straight line joining the points $\left(\frac{0.1}{T}, 0°\right)$ and $\left(\frac{10}{T}, \pm 90°\right)$ that passes through 45° at ω = 1/T. The maximum error between the exact plot and the straight line approximation is found to be within 5.5°. Fig. 4.19 and Fig. 4.20 show typical plots for $n = 1$ and $n = -1$ respectively.

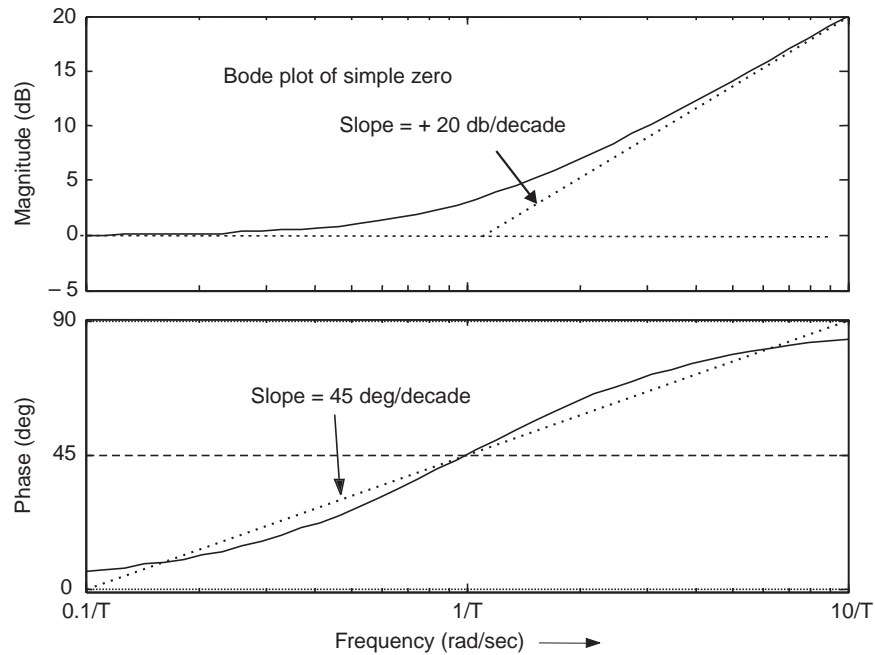



Fig. 4.19 Bode plot of simple zero, $F(j\omega) = (1 + j\omega T)$

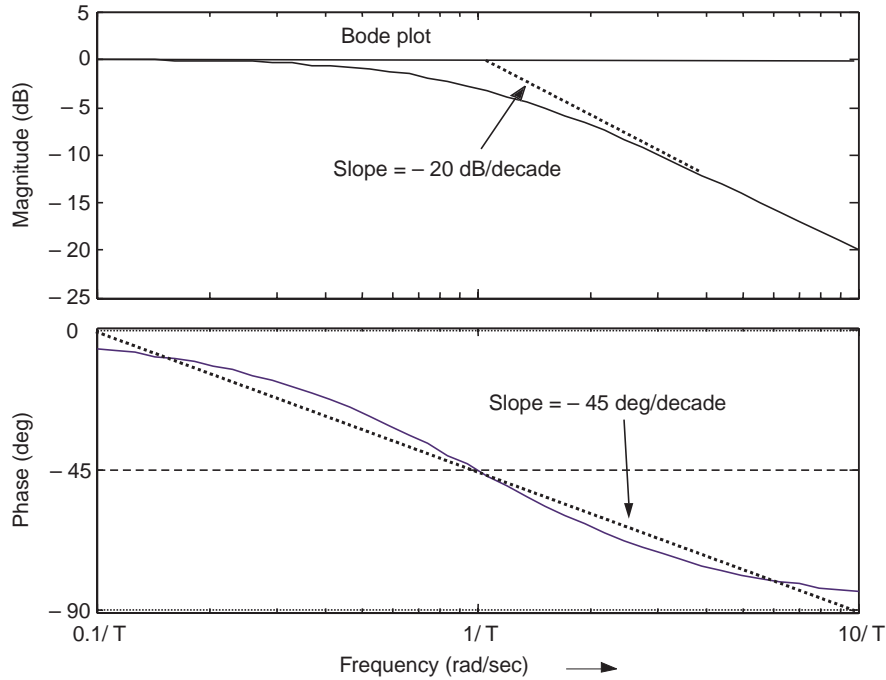



Fig. 4.20 Bode plot of simple pole, $F(j\omega) = (1 + j\omega T)^{-1}$

Bode plots are extensively used in the preliminary stages of analysis and design of a control system, since it gives the general nature of the frequency-response characteristics quickly and fairly accurately with a minimum amount of calculation. After the design is complete, the detailed simulation of the entire system is undertaken for final implementation.

(D) Factors corresponding to complex poles or zeros $[1 + 2\delta + (j\omega/\omega_n)^2] \pm 1$.

Quadratic factors of the following form are encountered in control systems :

$$F(j\omega) = \frac{1}{1 + 2\delta\left(\frac{j\omega}{\omega_n}\right) + \left(\frac{j\omega}{\omega_n}\right)^2} \tag{4.88}$$

The nature of roots in the above expression depends on the value of δ. For δ > 1, both the roots will be real and the quadratic factor can be expressed as a product of two first-order factors with real poles. The quadratic factor can be expressed as the product of two complex-conjugate factors for values of δ satisfying $0 < \delta < 1$. Asymptotic approximations to the frequency-response curves will not be accurate for the quadratic factor for small values of δ. This is because of the fact that the magnitude as well as the phase of the quadratic factor depend on both the corner frequency and the damping ratio δ.

The asymptotic frequency-response curve is given by

$$F_{dB} = 20 \log_{10} \left| \frac{1}{1 + 2\delta\left(j\frac{\omega}{\omega_n}\right) + \left(j\frac{\omega}{\omega_n}\right)^2} \right| \tag{4.89}$$

$$= -40\log_{10}\left(\frac{\omega}{\omega_n}\right), \quad \text{for } \omega >> \omega_n \tag{4.90}$$

$$= 0 \qquad\qquad \text{for } \omega << \omega_n$$

The frequency ω_n is the corner frequency for the quadratic factor considered.

The low frequency and high frequency asymptotes are independent of the value of δ (vide Fig. 4.16(*a*)). A resonant peak occurs near the frequency $\omega = \omega_n$, which is also revealed by the Equation (4.89). The magnitude of the resonant peak is determined by the damping ratio δ. Straight line approximation of the frequency response by asymptotes will be, obviously, erroneous. The magnitude of the error depends on the value of δ and it is large for small value of δ.

The phase angle of the quadratic factor $\left[1 + 2\delta\left(j\frac{\omega}{\omega_n}\right) + \left(j\frac{\omega}{\omega_n}\right)^2\right]^{-1}$ is given by

$$\angle F(j\omega) = -\tan^{-1}\left[\frac{2\delta\dfrac{\omega}{\omega_n}}{1-\left(\dfrac{\omega}{\omega_n}\right)^2}\right] \tag{4.91}$$

The phase angle is found to depend on both ω and δ. It is 0° at ω = 0 and – 90° at the corner frequency $\omega = \omega_n$, whereas the phase angle approaches – 180° as $\omega \to \infty$. The phase angle at the corner frequency is independent of δ, since

$$\angle F(j\omega) = -\tan^{-1}\left(\frac{2\delta}{0}\right) = -\tan^{-1}\infty = -90°$$

Even though asymptotic lines can not be used for amplitude and phase plots for quadratic factors, a template prepared with typical values of δ may be kept ready as help for a rough sketch.

Using MATLAB control Toolbox, the Bode plot between frequencies $w_{\min}$ and $w_{\max}$ of a transfer function F, expressed in terms of its poles and zeros, can be found by the command Bode [F,{$w_{\min}$, $w_{\max}$}].

For example, the Bode plot of the transfer function $F(s) = \dfrac{10(s+5)}{s(s+2)(s^2+5s+100)}$ can be obtained by the following :

```
% Script_Figure4.21
num1=[1 5]; num=10*num1; den=[1 5 100];
g1=tf(num,den);  g2=zpk([ ],[0 -2],1);
F=g1*g2; Bode(F,{0.01, 100})
```

The Bode plot is shown in Fig. 4.21.

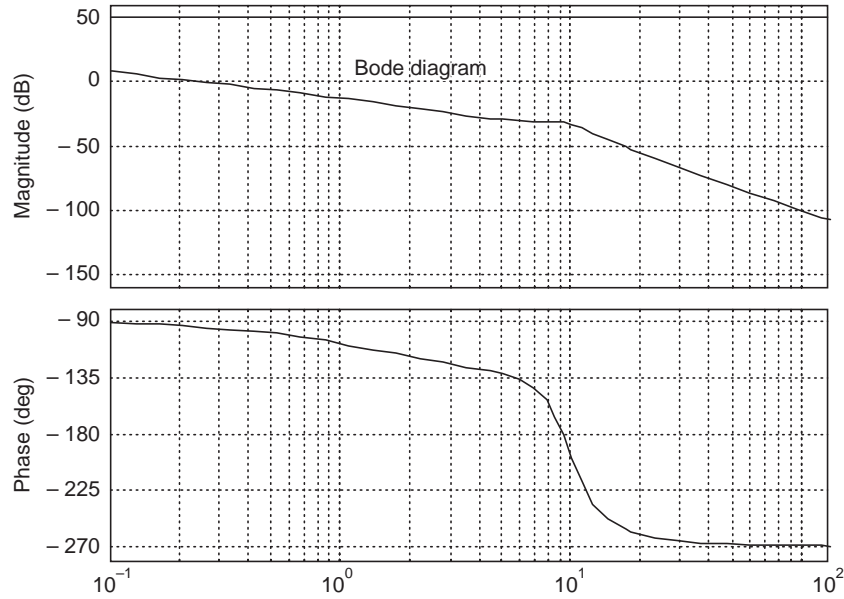



Fig. 4.21 Bode plot of $F(s) = \dfrac{10(s+5)}{s(s+2)(s^2+5s+100)}$

Chapter 4

The resonant frequency ω_r and the resonant peak value M_r: We have already noted in Section 4.11.1 that the magnitude of $F(j\omega)$ in (4.88) will attain a maximum value, M_r, given by relation (4.79) at a frequency $\omega_r = \omega_n\sqrt{1-2\delta^2}$

We have also noted that $\delta > 0.707$, $M_r < 1$ (4.92)

As δ approaches zero, M_r approaches infinity. This means that the system has an output with zero input, implying that the system is unstable and is useless as a control system. But it may be used as oscillators (signal generators) with natural frequency of oscillation ω_r.

The phase angle of $F(j\omega)$ at the frequency where the resonant peak occurs can be obtained by substituting Equation (4.78) into Equation (4.91). Thus, at the resonant frequency ω_r,with $\delta > 0$, we have

$$F(j\omega_r) = -\tan^{-1}\frac{\sqrt{1-2\delta^2}}{\delta} = -90° + \sin^{-1}\frac{\delta}{\sqrt{1-\delta^2}} \quad (4.93)$$

4.13 PROCEDURE FOR MANUAL PLOTTING OF BODE DIAGRAM

Using the guidelines discussed above, the Bode plot for minimum phase system can be readily obtained on a graph paper. The first step is to express $G(j\omega)H(j\omega)$ in factored form in terms of its time constants. If the product terms appear as poles and zeros, it should be converted to time constant form. That is, a factor $(s + 2)$ should be recast as $2(j0.5\omega + 1)$ for which the corner frequency is at $\omega = 2$. The magnitude and phase curves for individual factors should then be approximated by asymptotic lines. If desired, corrections to magnitude and phase curves may be introduced by comparing with the prototype curves. The final plot is obtained by graphically adding the individual plots.

4.14 MINIMUM PHASE AND NON-MINIMUM PHASE SYSTEMS

A transfer function with none of the poles and zeros lying on the right hand side of the imaginary axis in the s-plane is referred to as a minimum phase transfer function, whereas the transfer functions having poles and/or zeros in the right-half s-plane are non-minimum phase transfer functions. The corresponding systems are referred to as *minimum phase* systems and *non-minimum phase* systems, respectively.

One of the important properties of a minimum phase system is that its transfer function can be uniquely determined from the magnitude curve alone. For a non-minimum phase system, this can not be done.

As an illustrative example, let us consider two systems whose sinusoidal transfer functions are, respectively,

$$G_1(j\omega) = \frac{1 + j\omega T}{1 + j\omega T_1} \text{ and } G_2(j\omega) = \frac{1 - j\omega T}{1 + j\omega T_1} \; (0 < T < T_1)$$

The amplitude for both $|G_1(j\omega)|$ and $|G_2(j\omega)|$ are the same but the phase lag of $G_2(j\omega)$ is more than that of $G_1(j\omega)$.

Non-minimum phase systems may be encountered in two different situations. In one case the process includes elements which are non-minimum phase by nature like the transport lags whereas, in the other, one or more poles of an otherwise minimum phase system have moved to the unstable region giving rise to non-minimum phase character.

If m and n are, respectively, the degrees of the numerator and denominator polynomials of the transfer function, then the phase angle of a minimum phase system becomes $-90° (m - n)$ at $\omega = \infty$.

Non-minimum phase systems are sluggish in response due to excessive phase lag. So, if fast response is of primary consideration, one should avoid using non-minimum phase components. Transport lag is the most commonly encountered example of non-minimum phase element in control systems.

Transport lag. Transport lags normally exist in thermal, hydraulic, and pneumatic systems. Transport lag has excessive phase lag with no attenuation at high frequencies.

A transport lag may be represented as :

$$F(j\omega) = e^{-j\omega T}$$

The magnitude of $|F(j\omega)|$ is found to be unity since

$$|F(j\omega)| = |\cos \omega T - j \sin \omega T| = 1$$

Therefore, the log magnitude of the transport lag $e^{-j\omega T}$ is equal to 0 db. The phase angle of the transport lag is

$$\angle F(j\omega) = -\omega T = -57.3\, \omega T \text{ (degrees).}$$

The phase angle varies linearly with the frequency ω.

Example 4.1 Let us consider a transfer function with transport lag given by:

$$F(j\omega) = \frac{e^{-j\omega T_1}}{1 + j\omega T}$$

The log magnitude is

$$20 \log |F(j\omega)| = 20 \log_{10} |e^{-j\omega T_1}| + 20 \log_{10} \left| \frac{1}{1 + j\omega T} \right| = 20 \log_{10} \left| \frac{1}{1 + j\omega T} \right|$$

The phase angle of F($j\omega$) is given by :

$$F(j\omega) = \angle e^{-j\omega T1} + \angle \frac{1}{1+j\omega T} = -\omega T_1 - \tan^{-1}\omega T$$

The log-magnitude and phase-angle curves for this transfer function with $T_1 = T = 1$ sec is shown in Fig. 4.22.

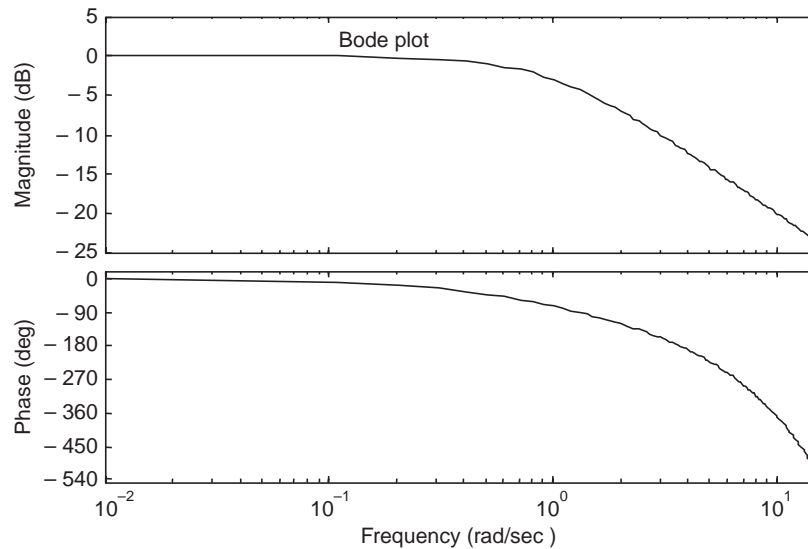



Fig. 4.22 Bode plot of $F(j\omega) = \frac{e^{-j\omega T_1}}{1+j\omega T}$

MATLAB Command: *F=zpk([],[-1],1,'iodelay',1) ; Bode(F,{0.1 20});*

MATLAB SCRIPTS

```
% Script_Fig4.7
% Plot of step response with damping ratio δ as parameter
clear all; close all hidden
% d= 2 δ ωn
d=[0 0.1 0.2 0.4 0.6 0.8 1 1.2 1.4 1.6 1.8 2 4]        % delta = 0, 0.05 ,1 etc
num=[1];                                                % ωn = 1
t=[0:0.1:12];
for i=1:13
d1=d(i)
den=[1 d1 1];                                           %Refer to Equation (4.6)
[y,x]=step(num,den,t);
Ys(:,i)=y;
end
plot(t,Ys(:,1),'-',t,Ys(:,2),'-',t,Ys(:,3),'-',t,Ys(:,4),'-',t,Ys(:,5),'-',t,Ys(:,6),'-',t,Ys(:,7),'-',t,...
Ys(:,8),'-',t,Ys(:,9),'-'t,Ys(:,10),'-',t,Ys(:,11),'-',t,Ys(:,12),'-',t,Ys(:,13),'-')
xlabel('Time(Sec)'ylabel('y(t)')
```

Chapter 4

```
%Script_Fig4.9
%Impulse response clear a second order prototype system
% d= 2*delta*omegan
d=[0.4 1.0 2 3.0]                    % delta = 0, 0.05 ,1 etc
num=[1];                             % omegan = 1
t=[0:0.1:12];
for i=1:4;  d1=d(i);                 %Refer to Equation (4.6)
den=[1 d1 1];
[y,x]=impulse(num,den,t);
Ys(:,i)=y;
end
plot(t,Ys(:,1),'-',t,Ys(:,2),'-',t,Ys(:,3),'-',t,Ys(:,4),'-')
xlabel('Time(Sec)'),ylabel('y(t)')
```

```
%Script_4.13
% Steady state error with parabolic input.
clear all; close all hidden
t=[0:0.01:6]; u=0.5*power(t,2);      % R= 1, Refer to equation (4.40)
num=[4]; den=[1 3.2 4];
[y,x]=l sim(num,den,u,t);
plot(t,y,t,u);                       %
xlabel('Time (sec)'), ylabel('y(t)')
```

```
% Script_Fig4.16a
% Plot M(jω) VS normalised frequency u with damping ratio δ as parameter
clear all; close all hidden
d=[ 0.1 0.2 0.4 0.6 0.8 1.0 2.0 4.0 8.0];
for i=1:9
u=[0.01:0.1:10]; num=[1];            % ωn = 1
d1=d(i); % d1 = 2 δ ωn ;
den=[1 d1 1];                        % Refer to Equation (4.72)
y=bode(num,den,u);
Ys(:,i)=y;
end
loglog(u,Ys(:,1),'-',u,Ys(:,2),'-',u,Ys(:,3),'-',u,Ys(:,4),'-',u,Ys(:,5),'-', u, ...
Ys(:,6),'-',u,Ys(:,7),'-',u,Ys(:,8),'-',u,Ys(:,9),'-')
xlabel('\omega'),ylabel('|M(j\omega)| in dB')
```

```
% Script_Fig4.17
% Plot of BW with damping ratio δ
clear all; close all hidden
d=0.0:0.01:1; L=length(d);
for i=1:L; d1=d(i);                  % damping ratio
% Refer to Equation (4.83)
```

```
d2=1-2*power(d1,2);
d3=4*power(d1,4)-4*power(d1,2)+2;
d4=power(d3,0.5); d5=d2+d4;
y=power(d5,0.5);
Ys(:,i)=y;
end
plot(d,Ys,'-')
xlabel('\delta'),ylabel('BW/\omegan')
```

```
% Script_ Fig4.18
% Amplitude and phase plot for poles and zeros located at the origin
clear all; close all hidden
num=[1 0];den=[1];sys=tf(num,den);bode(sys,'k');hold on;   % single zero at origin
num=[1 0 0];den=[1];sys=tf(num,den);bode(sys,'k',{.1,100});hold on;  % double zero
% at origin
num=[1 0 0 0];den=[1];sys=tf(num,den);bode(sys,'k');hold on; % tripple zero at origin
den=[1 0];num=[1];sys=tf(num,den);bode(sys,'k');hold on;   % single pole at origin
den=[1 0 0];num=[1];sys=tf(num,den);bode(sys,'k');hold on;  % double zero at origin
den=[1 0 0 0];num=[1];sys=tf(num,den);bode(sys,'k');   % tripple pole at origin
```

Chapter 4

REVIEW EXERCISE

RE4.1 Consider a unity feedback system with forward path transfer function G(s) given by :

$$G(s) = \frac{200}{(s+20)}$$

Find the bandwidth of the open-loop system and that of the closed-loop system and compare the results.

Ans.: open-loop BW = 20, closed-loop BW = 220.

RE4.2 Bode-diagram of a minimum phase open loop amplifier is shown in Fig. RE4.2.

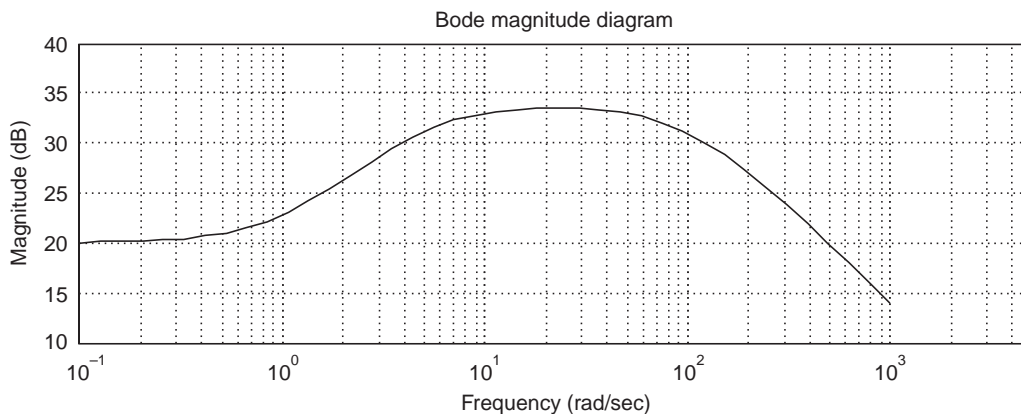



Fig. RE4.2 Bode plot of minimum-phase system

(*a*) Find the transfer function of the amplifier. **Ans.:** $G(s) = \dfrac{10(s+1)}{(0.2s+1)(0.01s+1)}$

(*b*) Find the expected percent overshoot for a step input for the closed-loop system with unity feedback, (*c*) estimate the bandwidth of the closed-loop system, and (*d*) the setting time (2% criterion) of the system.

Ans: (*b*) 7.73 % (*c*) 5857 rad/sec (*d*) t_s = 1.26 sec

RE4.3 The output and input of a position control system is related by:

$$Y(s) = \frac{500(s+100)}{s^2 + 60s + 500} R(s).$$

(*a*) If $r(t)$ is a unit step input, find the output $y(t)$.

(*b*) What is the final value of $y(t)$?

Ans: (*a*) $y(t) = 100 - 112.5e^{-10t} + 12.5e^{-50t}$, (*b*) $y_{ss} = 100$

PROBLEMS

P4.1 The pure time delay e^{-sT} may be approximated by a transfer function as:

$$e^{-sT} \cong \frac{(1 - Ts/2)}{(1 + Ts/2)}$$

for 0 < ω < 2/T. Obtain the Bode diagram for the actual transfer function and the approximation for T = 2 for 0 < ω < 1.

Hints: MATLAB command < g = zpk([ ],[ ],1,'inputdelay', 2)> will produce transfer function $g = e^{-2s}$.

P4.2 The pneumatic actuator of a tendon-operated robotic hand can be represented by

$$G(s) = \frac{2500}{(s+45)(s+340)}$$

(*a*) Plot the frequency response of $G(j\omega)$ and show that the magnitudes of $G(j\omega)$ are – 16 dB and – 34.8 dB respectively at ω = 10 and ω = 300. Also show that the phase is –141° at ω = 500.

P4.3 The dynamics of a vertical takeoff aircraft are approximately represented by the transfer function $G(s) = \dfrac{8}{(s^2 + 0.25)}$

The controller transfer function is represented by (see Fig. P4.3)

$$G_c(s) = \frac{K_1(s+6)}{(s+2)} \text{ and } H(s) = s$$

(*a*) Obtain the Bode plot of the loop transfer function $L(s) = G_c(s)G(s)H(s)$ with $K_1 = 1.5$. (*b*) Compute the steady-state error for the closed loop system for a wind disturbance of $D(s) = 1/s$. (*c*) Obtain the frequency response $L(j\omega)$ and the peak amplitude of the resonant peak along with the resonance frequency (*d*) Find out the gain and phase margins from $L(j\omega)$ plot (*e*) Estimate the damping ratio of the system from the phase margin.

Ans: (*b*) e_{ss} = 1(*c*) M_r= 324 dB, ω_r= 0.5, (*d*) GM = ∞ , PM = 74.04° at ω_{cp} = 13.07 rad/s (*e*) δ = 0.74

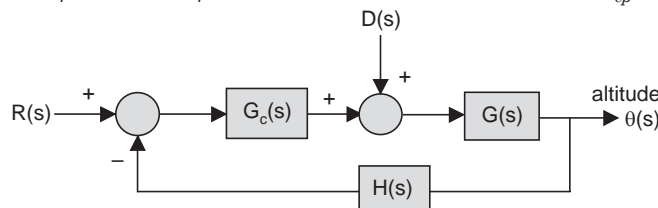



Fig. P4.3

P4.4 A system is described by a set of differential equations as shown below:

$$\dot{y}(t) = 2y(t) + a_1 x(t) = 2u(t)$$

$$\dot{x}(t) - a_2 y(t) = -4u(t)$$

where $u(t)$ is an input. (*a*) Select a suitable set of state variables and obtain the state variable representation (*b*) Find the characteristic roots of the system in terms of the parameters a_1 and a_2.

Ans: (*b*) $s = -1 \pm \sqrt{1 - a_1 a_2}$

P4.5 The simplified state variable vector representation for depth control of a submarine is given by :

$$\dot{\mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 \\ -0.0075 & -0.1115 & 0.125 \\ 0 & 0.075 & -0.265 \end{bmatrix} x + \begin{bmatrix} 0 \\ -0.1 \\ +0.075 \end{bmatrix} \mathbf{u}(t)$$

where $\mathbf{u}(t)$ is the deflection of the stern plane. (*a*) Examine the stability of the system. (*b*) Obtain the discrete-time approximation with sampling period of 0.25 sec as well as 2.5 sec. Obtain and compare the responses for both the sampling periods.

Ans: (*b*) T = 0.25,

$$x(k+1) = \begin{bmatrix} 0.99977 & 0.24655 & 0.00379 \\ -0.00185 & 0.97256 & 0.02982 \\ -0.00002 & 0.01789 & 0.93677 \end{bmatrix} x(k) + \begin{bmatrix} -0.00307 \\ -0.02437 \\ 0.01792 \end{bmatrix} u(k)$$

T = 2.5,

$$x(k+1) = \begin{bmatrix} 0.97859 & 2.18290 & 0.28817 \\ -0.01637 & 0.75681 & 0.19649 \\ -0.00130 & 0.11790 & 0.53281 \end{bmatrix} x(k) + \begin{bmatrix} -0.26605 \\ -0.19667 \\ 0.12105 \end{bmatrix} u(k)$$

P4.6 The forward path transfer function, with unity negative feedback, is given by :

$$G(s) = \frac{10(s+4)}{s(0.1s+1)(2s+1)}$$

(*a*) Find the steady state error due to ramp input. (*b*) Find the dominant roots of the closed loop system and estimate the settling time (2% criteria) to a step input. (*c*) Compute the step response and find over shoot and settling time and compare with results in part (*b*).

P4.7 Machine tools are automatically controlled as shown in Fig. 4.7. These automatic systems are called numerical machine controls. Considering one axis, the desired position of the machine tool is compared with the actual position and is used to actuate a solenoid coil and the shaft of a hydraulic actuator. The transfer function of the actuator is given by:

$$G_1(s) = \frac{X(s)}{Y(s)} = \frac{1}{s(0.5s+1)}$$

The output voltage of the difference amplifier is

$$E_0(s) = A[X(s) - R_d(s)],$$

where $r_d(t)$ is the desired position input. The force on the shaft is proportional to the current i so that $F = K_2 i(t)$, where $K_2 = 2.0$. The force is balanced against the spring. $F = -Ky(t)$, where K is the spring constant and is numerically equal to 1.2, and R = 10 ohm, and L = 0.5 henry.

(*a*) Determine the forward path transfer function and the gain A such that the phase margin is 50°. (*b*) For the gain A of part (*a*), determine the resonance peak M_r, resonance frequency ω_r and the bandwidth of the closed-loop system. (*c*) Estimate the percent overshoot of the transient response for a unity step of the desired position and the settling time to within 2% of the final value.

Ans: (*a*) $G(s) = \dfrac{2.4\,A}{s\,(0.5s + 10)\,(0.5s + 1)}$, A = 6.95

(*b*) M_r = 1.23 db, ω_r = 1.29 rad/sec, BW = 2.31 rad/sec

(*c*) 16.2%,t_s = 4.49 sec

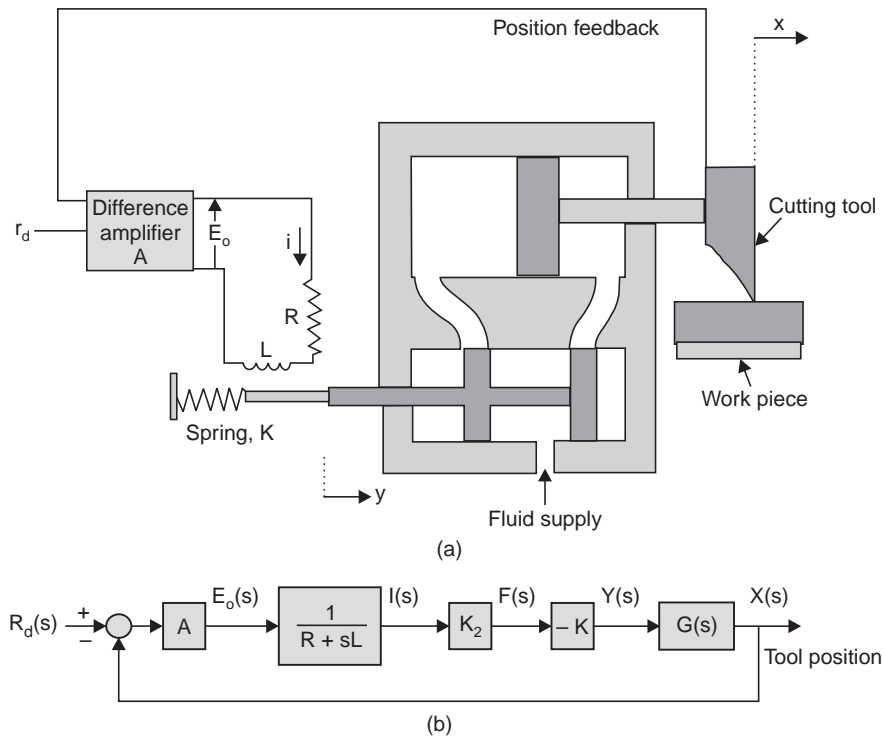



Fig. P4.7 (a) Tool position control, (b) Block diagram

P4.8 A closed loop system for controlling chemical concentration is shown in Fig. P4.8. The feed is granular in nature and is of varying composition. It is desired to maintain a constant composition of the output mixture by adjusting the feed-flow valve. The transfer function of the tank and output valve is given by

$$G(s) = \frac{4}{4s + 1}$$

and that of the controller is represented as :

$$G_c(s) = K_1 + \frac{K_2}{s}$$

The transport of the feed along the conveyor introduces a delay time of 2 seconds. (*a*) Draw the Bode diagram when $K_1 = 1$, $K_2 = 0.2$, and investigate the stability of the system. (*b*) Also draw the Bode diagram when $K_1 = 0.1$ and $K_2 = 0.02$, and investigate the stability of the system.

Ans.: (*a*) unstable (*b*) stable, GM = 20.5 dB PM = 84.7°.

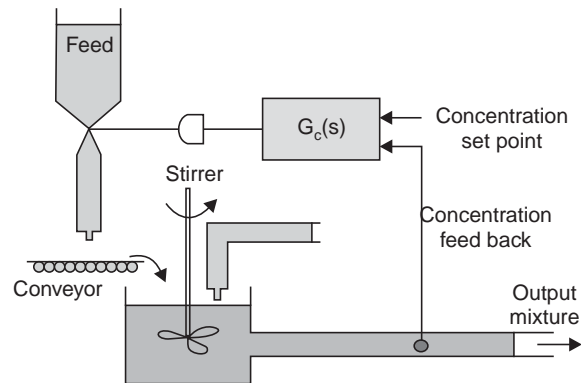



Fig. P4.8 Chemical concentration control

P4.9 In an automatic ship-steering system the deviation of the heading of the ship from the straight course is measured by radar and is used to generate the error signal. The block diagram representation shown in Fig. P4.9. This error signal is used to control the rudder angle $\delta(s)$. The transfer function of the ship-steering system is given by:

$$G(s) = \frac{E(s)}{\delta(s)} = \frac{0.16\,(s + 0.18)\,(-s + 0.3)}{s^2(s + 0.24)\,(s + 0.3)}$$

where $E(s)$ is the Laplace transform of the deviation of the ship from the desired heading and $\delta(s)$ is the transfer function of deflection of the steering rudder. Obtain the frequency response with $k_1 = 1$ and $k_2 = 0$.

(*a*) Is this system stable with $k_1 = 1$ and $k_2 = 0$.? **Ans:** No

(*b*) With $k_1 = 1$ and $k_2 = 0$, is it possible to stabilize this system by lowering the forward path gain of the transfer function $G(s)$? **Ans:** No

(*c*) Repeat parts (*a*) when $k_1 = 0.01$ and $k_2 = 1$. **Ans:** yes

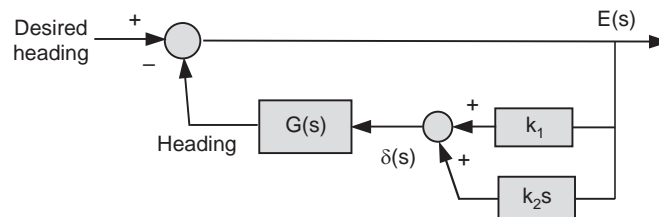



Fig. P4.9 Automatic ship steering

P4.10 A typical chemical reactor control scheme is shown in Fig. P4.10. The chemical process is represented by G_3 and G_4 and disturbance by $D(s)$. The controller and the actuator valve are represented by G_1 and G_2 respectively and the feedback sensor is represented by $H(s)$. We will assume that G_2, G_3, and G_4 are all of the form:

$$G_i(s) = \frac{K_i}{1 + \tau_i s}$$

where $\tau_3 = \tau_4 = 5$ seconds, and $K_3 = K_4 = 0.2$ and $H(s) = 1$. The valve constants are $K_2 = 10$ and $\tau_2 = 0.4$, second. The close loop system is required to maintain a prescribed steady-state error.

(*a*) With $G_1(s) = K_1$, find the proportional gain such that steady state error is less than 5% of the step input. For this value of K_1, find the overshoot to a step change in the reference signal $r(t)$.

(*b*) Now with a PI controller $G_1(s) = K_1(1 + 1/s)$, find K_1 so as to get an overshoot less than 25% but greater than 5%. For these calculations $D(s)$ is assumed to be zero.

Chapter 4

(c) Now with $r(t) = 0$, find the settling time of the output within 2% of the steady state value for cases (a) and (b) when subjected to a step disturbance .

Ans: (a) $K_1 \geq 47.5$, with $K_1 = 50$, over shoot 78.9% (b) $K_1 = 0.2$, over shoot = 14.7%.
(c) $t_s = 73.4$ sec, and 92.1 sec

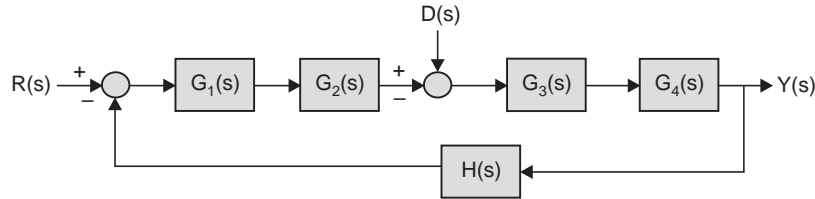



Fig. P4.10 Chemical reactor control

P4.11 Consider a system with a closed-loop transfer function

$$M(s) = \frac{Y(s)}{R(s)} = \frac{1.44}{(s^2 + 0.6s + 0.36)(s^2 + 0.16s + 4)}$$

(a) Plot the frequency response and step response.

P4.12 The Bode magnitude plot of transfer function

$$G(s) = \frac{K(0.2s + 1)(\tau_1 s + 1)}{s(0.1s + 1)(0.05s + 1)(\tau_2 s + 1)}$$

is shown in Fig. P4.12. Determine K, τ_1 and τ_2 from the plot.

Ans: $K = 4$, $\tau_1 = 1$, $\tau_2 = 1/50$

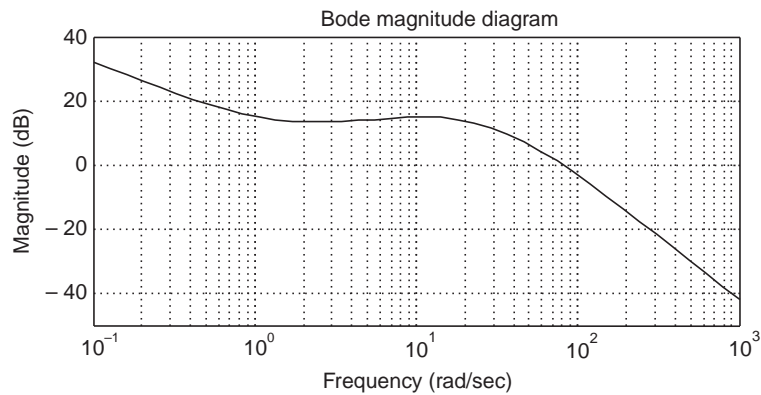



Fig. P4.12 Bode plot of G(s)

CHAPTER

5

The Stability of Linear Control Systems

5.1 THE CONCEPT OF STABILITY

The concept of stability of systems is of paramount importance and from an application point of view an unstable system is of no use. One of the most important tasks a control engineer is required to do, in connection with the design and analysis of a control system, is to ensure its stability. As with many other general statements, there are exceptions, but in the present context, it may be stated that all the control designs must result in a closed-loop stable system. Many physical systems are inherently open-loop unstable, and some systems are even designed intentionally to be open-loop unstable. Most modern fighter aircraft are open-loop *unstable by design,* and the pilot cannot fly it without stabilization with feedback control. Feedback not only stabilizes an unstable plants, it also takes care of the transient performance by a judicious choice of controller parameters. The performance specifications may take the form of steady-state tracking errors, percent overshoot, settling time, time to peak, and the other indices already discussed in Chapters 4 and a few that will be considered in Chapter 11.

A stable system may be defined as a dynamic system which produces a bounded output when the corresponding input is bounded. That is, if the system is subjected to a bounded input or disturbance and the response is bounded in magnitude, the system is said to be stable. This kind of stability is known as bounded-input, bounded-output stability.

In order to illustrate the concept of stability, let us consider the *three possible resting positions (equilibrium states)* of a right circular cone as shown in Fig. 5.1. When it is resting on its base on a horizontal plane surface (Fig. 5.1(*a*)) and is tilted slightly from its vertical position,

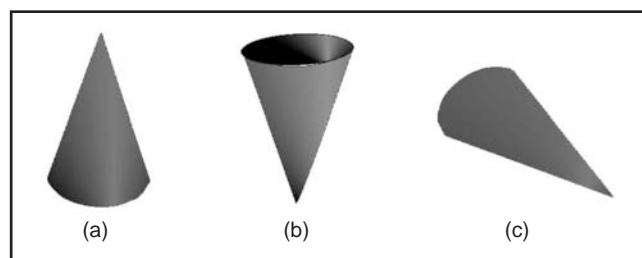



Fig. 5.1 Equilibrium states of a cone (a) Stable (b) Unstable and (c) Neutrally stable equilibrium

it will respond by returning to its original vertical position. So, the vertical position is a stable equilibrium position. On the other hand, if we try to make the cone stand on its tip (Fig. 5.1(*b*)),

which might need the skill of a magician or a blunt tip, it falls immediately to its side as soon as we withdraw the support or with a puff of wind. So, this position is an unstable position. Now, if the cone is placed on its side (Fig. 5.1(*c*)) and slightly pushed, it will roll on its side but still rests on its side position. This side position is said to be the neutrally stable position.

We can define the stability of a dynamic systems in an analogous way. A dynamic system that returns to its equilibrium state following a slight perturbation is referred to as a stable system. If it moves away and does not return to its original equilibrium state, it will be referred to as an unstable system. And if it responds to the perturbation but does not move away from its initial equilibrium state, it will be referred to as a neutrally stable system.

So a system is said to be a stable system if its response is bounded with a bounded (limited) input.

It has been observed in Chapter 4 that the response of a linear system depends on the roots of the closed loop system. Therefore, the stability of a system can be ultimately related to the nature of the roots of the characteristic equation of a closed loop transfer function. The general form of transfer function of an nth -order closed-loop system with real poles and complex conjugate poles can be expressed as:

$$\frac{Y(s)}{U(s)} = M(s) = \frac{K \prod_{i=1}^{q} (s + z_i)}{\prod_{k=1}^{r} (s + \sigma_k) \prod_{j=1}^{p} [s^2 + 2\alpha_j s + (\alpha_j^2 + \omega_j^2)]} = \frac{P(s)}{Q(s)}$$

where $n = r + 2p \geq q$. The characteristic equation is obtained by setting the denominator of M(s) to zero whose roots are the poles of the closed loop system. The unity impulse response $g(t)$ for the output is of the form

$$g(t) = \sum_{k=1}^{r} A_k e^{-\sigma_k t} + \sum_{j=1}^{p} B_j \left(\frac{1}{\omega_j}\right) e^{-\alpha_j t} \sin(\omega_j t + \theta_j) \qquad (5.1)$$

where A_k and B_j are constants and computed in terms of σ_k, z_i, α_j, K and ω_j. If σ_k and α_j are positive, the contribution of the exponential terms will die out with time t and impulse response $g(t)$ will be bounded. Therefore, if the roots of the characteristics equation lie on the left half side of the imaginary axis in the s-plane the impulse response will be finite and bounded. Hence, the *necessary and sufficient condition for stability of a system is that all the roots of the characteristic equation have negative real parts***.** If any of the roots has positive real part, the contribution from the corresponding exponential term will grow with time and the impulse response will be unbounded and the entire system will be referred to as unstable. Now if $\sigma_k > 0$ for all k and $\alpha_j = 0$ for one j (*i.e.,* one pair of roots on the imaginary axis), we see from relation (5.1) that one sinusoidal term in the impulse response is sustained as $t \to \infty$, but it does not grow with time. In such a case the system is referred to as neutrally (or marginally) stable. However, if there are repeated roots on the imaginary axis, a term of the form $t(\sin \omega_j t + \theta_j)$ will appear in the expression of impulse response (5.1) and the response will be unbounded as $t \to \infty$. The output of a marginally stable system will be unbounded due to resonance, when it is excited with a finite-amplitude sinusoidal input whose frequency is equal to the imaginary axis pole.

The system with characteristic equation given by $(s + 2)(s^2 + 25) = 0$ is marginally stable. If this system is excited by a sinusoid of frequency $\omega = 5$, the output becomes

unbounded. In any case, the marginally stable system is useless for any control application other than when it is used as a signal generator.

Let us now examine the total solution of the state equation from Chapter 3 repeated below:

$$\mathbf{x}(t) = e^{a(t-t_0)}\,\mathbf{x}(t_0) + \int_{t_o}^{t} e^{-at}\,\mathbf{bu}\,(t-\tau)d\tau \tag{5.2}$$

$$= \mathbf{g}(t-t_0)\,\mathbf{x}(t_0) + \int_{t_0}^{t} \mathbf{g}(t)\,\mathbf{bu}\,(t-\tau)d\tau \tag{5.3}$$

where $g(t)$ is the impulse response. When the impulse response $g(t)$ is bounded and satisfies the additional condition

$$\lim_{t\to\infty} \;|\; g(t) \;|\; \to 0 \tag{5.4}$$

then the integral in relation (5.3) will be finite for a bounded input $|u(t)|$ and a matrix $\mathbf{b}$ with finite elements.

Taking the absolute value on both sides of Equation (5.3)

$$|\;\mathbf{x}(t)\;| = |\;\mathbf{g}(t-t_0)\;|\;|\;\mathbf{x}(t_0)\;| + \left|\;\int_{t_o}^{t} \mathbf{g}(t)\,\mathbf{bu}(t-\tau)d\tau\;\right|$$

$$\Rightarrow \quad |\;\mathbf{x}(t)\;| \le |\;\mathbf{g}(t-t_0)\;|\;|\;\mathbf{x}(t_0)\;| + \int_{t_o}^{t} |\;\mathbf{g}(t)\,\mathbf{b}\;|\;|\;\mathbf{u}(t-\tau)\;|\;d\tau$$

$$\Rightarrow \quad |\;x(t)\;| \le \mathrm{M}_1 + \mathrm{M}_2 = \mathrm{M}$$

where $\mathrm{M}_1 = |\;\mathbf{g}(t-t_0)\;|\;|\;\mathbf{x}(t_0)\;|$ and $\mathrm{M}_2 = \int_{t_o}^{t} |\;\mathbf{g}(t)\,\mathbf{b}\;|\;|\;\mathbf{u}(t-\tau)\;|\;d\tau$

Therefore, the response $|\mathbf{x}(t)|$ is bounded when the initial conditions $\mathbf{x}(t_0)$ and the input $\mathbf{u}(t)$ are bounded, provided the impulse response is finite and approaches zero as t approaches to infinity. Since $\mathbf{y}(t) = \mathrm{C}\mathbf{x}(t) + \mathrm{D}\mathbf{u}(t)$, the system output will also be finite and bounded under the same conditions as mentioned above.

A stable closed-loop system, can be further characterized by the degree of stability. This is referred to as **relative stability.** The aircraft designers are very much aware of the notion of relative stability. The fighter aircrafts are designed to be less stable than the commercial transport aircrafts. This is the requirement of the fighter planes- it should be highly maneuverable. It is more difficult to maneuver a commercial aircraft very quickly since it is more stable thereby saving the passengers from violent turns and spins.

The absolute stability of a linear system is examined by determining that all the poles of the transfer function lie in the left half s-plane, or equivalently that all the eigenvalues of the system matrix A lie in the left half s-plane. Once it is established that all the poles (or eigenvalues) are in the left half s-plane, the relative stability is investigated by examining the relative locations of the closed loop poles (or eigenvalues).

The stability of a linear closed loop system can be determined by finding the roots of the characteristic equation $\mathrm{Q}(s) = 0$. However, it is advantageous to be able to know the stability-status of a system without solving the characteristic equation. As a matter of fact, several methods are available in the literature that can provide the answer regarding the status of absolute stability without solving the characteristic equation. These methods are based on the following three approaches:

(1) the s-plane approach,

(2) the frequency plane (jω) approach, and

(3) the time-domain approach.

We shall consider these approaches one by one.

The stability of a feedback system is found to be directly related to the location of the roots of the characteristic equation of the system. The Routh-Hurwitz technique is a useful tool for assessing the nature of the roots of a characteristic equation without solving it explicitly. The technique allows us to compute the number of roots of the characteristic equation that lie on the right hand side of the imaginary axis of the s-plane. This can be exploited as a design tool for determining the values of some system parameters to ensure the stability of the closed loop system. Once the stability is assured, the question of relative stability may be taken up to find the degree of stability of the closed loop system.

5.2 THE ROUTH-HURWITZ STABILITY CRITERION

A. Hurwitz and E.J. Routh independently published in 1890s, a method of investigating the stability of a linear system [16-17]. The Routh-Hurwitz stability method provides an answer to the question of stability by considering the characteristic equation of the system. The characteristic equation in the s-domain is written as

$$Q(s) = a_0 s^n + a_1 s^{n-1} + \ldots + a_{n-1} s + a_n = 0 \tag{5.5}$$

In the above equation, it is assumed that $a_n \neq 0$, *i.e.,* any root at $s = 0$ has been removed. We are interested to find out if any one of the roots of $Q(s)$ lies in the right half of the s-plane or not. Writing Equation (5.5) in factored form, we have

$$a_o (s - \alpha_1)(s - \alpha_2) \ldots (s - \alpha_n) = 0 \tag{5.6}$$

where $\alpha_i = i^{\text{th}}$ root of the characteristic equation. After multiplying the factors on the left hand side of the above equation, we get

$$\begin{aligned} Q(s) &= a_0 s^n - a_0(\alpha_1 + \alpha_2 + \ldots + \alpha_n)s^{n-1} + a_0(\alpha_1\alpha_2 + \alpha_2\alpha_3 + \alpha_1\alpha_3 + \ldots)s^{n-2} \\ &\quad - a_0(\alpha_1\alpha_2\alpha_3 + \alpha_1\alpha_2\alpha_4 \ldots)s^{n-3} + \ldots + a_0(-1)^n \alpha_1\alpha_2\alpha_3 \ldots \alpha_n = 0. \qquad (5.7) \\ &= a_o s^n - a_o \text{ (sum of all the roots)} s^{n-1} \\ &\quad + a_o \text{ (sum of the products of the roots taken 2 at a time)} s^{n-2} \\ &\quad - a_o \text{ (sum of the products of the roots taken 3 at a time)} s^{n-3} \\ &\quad + \ldots + a_o (-1)^n \text{ (product of all } n \text{ roots)} = 0. \qquad (5.8) \end{aligned}$$

From the relation (5.8), we get the following two necessary conditions that none of the roots have positive real parts.

(*i*) *all the coefficients in the above polynomial are of the same sign*

(*ii*) *all the coefficients are nonzero*

These conditions are necessary but not sufficient, which means that, if these conditions are not satisfied; we shall immediately conclude that the system is unstable; however, even if these conditions are satisfied, the stability is not guaranteed. For instance, let us consider the characteristic equation of an unstable system given by

$$q(s) = (s + 3)(s^2 - 2s + 10) = (s^3 + s^2 + 4s + 30)$$

Even though the system is unstable, yet all the coefficients of the polynomial are nonzero and positive.

The **Routh-Hurwitz criterion** is both necessary and sufficient condition for the stability of linear systems. The method was originally presented in terms of determinants formed with the elements of characteristic equation but we shall use the array formulation in a tabular form which is more convenient.

Let us consider characteristic equation of an nth order system given by:

$$a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + \ldots + a_{n-1} s + a_n = 0 \tag{5.9}$$

The Routh-Hurwitz criterion of stability will be stated in terms of the coefficients of the characteristic equation arranged in a tabular form as shown below:

The first row of the table is formed by writing the alternate coefficients of the characteristic equation starting with the coefficient of the highest power, s^n. Similarly, the second row is formed by writing the alternate coefficients of the characteristic equation starting with the coefficient of s^{n-1}.

Table 5.1

s^n	a_0	a_2	a_4	a_6 ......
s^{n-1}	a_1	a_3	a_5	a_7 ......
s^{n-2}	b_1	b_2	b_3	b_4 ......
s^{n-3}	c_1	c_2	c_3	c_4 ......
s^{n-4}	d_1	d_2	d_3	d_4 ......
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
s^2	e_1	e_2	0	0
s^1	f_1	0	0	0
s^0	g_1	0	0	0

The coefficients b_1, b_2, b_3, etc., are evaluated, in terms of the elements from the first two rows, as indicated below:

$$b_1 = \frac{a_1a_2 - a_0a_3}{a_1}$$

$$b_2 = \frac{a_1a_4 - a_0a_5}{a_1}$$

$$b_3 = \frac{a_1a_6 - a_0a_7}{a_1}$$

and so on.

The computation of the elements in the third row is continued until all the elements become zero. The same procedure is followed in evaluating the entries designated by c's, d's, e's, and so on in terms of the elements of the previous two rows. That is,

$$c_1 = \frac{b_1a_3 - a_1b_2}{b_1}$$

$$c_2 = \frac{b_1a_5 - a_1b_3}{b_1}$$

$$c_3 = \frac{b_1a_7 - a_1b_4}{b_1}$$

$$d_1 = \frac{c_1b_2 - b_1c_2}{c_1}$$

$$d_2 = \frac{c_1b_3 - b_1c_3}{c_1}$$

$$d_3 = \frac{c_1b_4 - b_1c_4}{c_1}$$

This process is continued until the *n*th row corresponding to s^o has been completed. It may be noted that in preparing the array an entire row may be *divided or multiplied* by a positive number in order to simplify the subsequent numerical calculation without affecting the stability condition (vide Example 5.2).

Routh's stability criterion states that the number of roots of Equation (5.9) with positive real parts is equal to the number of changes in sign of the coefficients in the first column of the array in the table. The criteria implies that for a system to be stable there should not be any changes in sign in the entries in the first column of the table. This is a both necessary and sufficient condition for ensuring stability. Since, the signs of the entries in the first column of the table influence the decision about stability, the exact values of the terms are not important. The necessary and sufficient condition that all roots of Equation (5.9) lie in the left-half s plane is that all the coefficients of Equation (5.9) be positive and all terms in the first column of the array are positive.

While preparing the table, we may encounter four different cases which require some modifications of the procedure for computing the elements of the array. The distinct cases are:

(1) *None* of elements in the first column is *zero*;

(2) a *zero* in the first column, but *other elements* in the row containing the zero are *nonzero*;

(3) a *zero* in the first column and the *other elements* of the row containing the zero are also *zero*, and

(4) as in case (3) plus *repeated roots* on the *jω-axis*.

In presence of the cases mentioned above, we will face difficulties in completing the table. We shall consider examples to illustrate the procedure for overcoming the difficulties in completing the array.

Case 1: None of the elements in the first column are zeros.

Example 5.1 Let us apply Routh's stability criterion to the following third-order polynomial:

$$a_0 s^3 + a_1 s^2 + a_2 s + a_3 = 0 \tag{5.10}$$

where all the coefficients are positive numbers. The Routh array becomes

s^3	a_0	a_2	0	← Padded zeros
s^2	a_1	a_3	0	
s^1	$\dfrac{a_1a_2 - a_0a_3}{a_1}$	0		
s^0	a_3	0		

We may add zeros padded to the right of the first two rows as shown in the table above for convenience of completing the entries in all the columns of the table. Very often this is implied and are not written explicitly. The zeros padded at the end of the first and second rows of the table have no bearing on the inference about stability.

The conditions that all roots of the characteristic equation (5.10) have negative real parts are found from the elements of first column of the above Table as : $a_i > 0$, $i = 0, 1, 2, 3$ together with $a_1a_2 > a_0a_3$

Example 5.2 Consider the following polynomial:

$$s^4 + 2s^3 + 4s^2 + 6s + 8 = 0 \tag{5.11}$$

Let us follow the procedure presented above and construct the array of coefficients.

s^4	1	4	8		s^4	1	4	8	
s^3	2	6	0		s^3	1	3	0	The second row is divided by 2
s^2	1	8	0	$\Rightarrow$	s^2	1	8	0	
s^1	– 10	0	0		s^1	– 5	0	0	
s_0	8	0	0		s_0	8	0	0	

In this example, the number of changes in sign of the coefficients in the first column is two: one change of sign from 1 to – 5 and another from – 5 to 8. This means that there are two roots with positive real parts. Note that the result is unchanged when the coefficients of any row are multiplied or divided by a positive number in order to simplify the computation.

Case 2: Zero in the first column of a row of which the other elements are nonzero.

If a term in the first column in any row is zero, but the remaining terms are nonzero, then the zero term is replaced by a very small positive number ε and the rest of the array is evaluated. As an example, consider the following equation :

$$s^5 + 3s^4 + 3s^3 + 9s^2 + 20s + 15 = 0 \tag{5.12}$$

The array of coefficients is

s^5	1	3	20
s^4	3	9	15
s^3	$0 \approx \varepsilon$	15	0
s^2	a	15	0
s^1	b	0	0
s^o	15	0	0

Here $a = (9\varepsilon - 45)/\varepsilon = -45/\varepsilon$ = a large negative number; and $b = (15a - 15\varepsilon)/a \approx 15$. There are two changes of sign. Therefore, there are two roots in the RHS of the imaginary axis in the s-plane. As a matter of fact the five roots of Equation (5.12) are at $s = -2.3326$, $s = 0.8603 \pm j\,1.7193$ and $s = -1.1940 \pm j\,0.6559$.

Case 3: Zero in the first column of a row of which other elements are also zero.

This case occurs when all the elements in one row are zero or when the row consists of a single element that is zero. This happens when the polynomial contain singularities that are symmetrical with respect to the origin in the s-plane. That is, it contains factors of the form $(s + \sigma)(s - \sigma)$, $(s \pm j\omega)$ or $(s - \sigma \pm j\omega)(s + \sigma \pm j\omega)$. In such a case, the evaluation of the rest of the array can be continued by forming an auxiliary polynomial with the coefficients of the last row and by using the coefficients of the derivative of this polynomial in the next row. Such roots that are symmetrical with respect to the origin in the s plane can be found by solving the auxiliary polynomial, which is always even. For a $2n$-degree auxiliary polynomial, there are n pairs of equal and opposite roots. For example, consider the following equation :

$$s^5 + s^4 - 6s^2 - 6s^2 + 25s + 25\ ;$$

The array of coefficients is

s^5	1	– 6	25	
s^4	1	– 6	25	← Auxiliary polynomial P(s)
s^3	0	0	0	

Chapter 5

The terms in the s^3 row are all zero. The auxiliary polynomial is then formed from the coefficients of the s^4 row. The auxiliary polynomial F(s) is

$$F(s) = s^4 - 6s^2 + 25$$

which indicates that there are two pairs of roots of equal magnitude and opposite sign. These pairs are obtained by solving the auxiliary polynomial equation F(s) = 0. The derivative of F(s) with respect to s is

$$\frac{dF(s)}{ds} = 4s^3 - 12s$$

The terms in the s^3 row are replaced by the coefficients of the last equation, that is, 4 and – 12. The array of coefficients then becomes

s^5	1	– 6	25	
s^4	1	– 6	25	
s^3	4	– 12	0	← Coefficients of dF(s)/ds
s^2	– 3	25	0	
s^1	64/3	0	0	
s^0	25	0	0	

We see that there are two changes in sign in the first column of the new array. Thus, the original equation has two roots with positive real parts. By solving for roots of the auxiliary polynomial equation,

$$s^4 - 6s^2 + 25 = (s^2 + 5)^2 - 16s^2 = (s^2 + 5 + 4s)\,(s^2 + 5 - 4s)$$

we obtain $\quad s = -2 \pm j,\ s = 2 \pm j$

These two pairs of roots are a part of the roots of the original equation. As a matter of fact, the original equation can be written in factored form as follows:

$$(s + 2 + j)\,(s + 2 - j)\,(s - 2 + j)\,(s - 2 - j)\,(s + 1) = 0$$

Clearly, the original equation has two roots with positive real parts.

Case 4: Repeated roots of the characteristics equation on the imaginary axis.

If the $j\omega$-axis roots of the characteristics equation are simple, it is neither stable nor unstable, it is marginally stable and its response contains an undamped sinusoidal mode. However, if the $j\omega$-axis has more than a pair of roots, the impulse response will be of the form tsin ($\omega t + \varphi$) and is unbounded as time approaches infinity. Routh-Hurwitz criteria *will not reveal* this type of instability.

Consider the transfer function: $\quad s^5 + 4s^4 + 5s^3 + 20s^2 + 4s + 16$

The array of coefficients is

s^5	1	5	4	
s^4	4	20	16	← Auxiliary polynomial P(s)
s^3	ε	ε	0	
s^2	16	16	0	
s^1	ε	0	0	
s^0	16	0	0	

where $\varepsilon \to 0$. It is to be noted that the absence of any sign-change misleads to the conclusion about the marginally stable nature of the system.

The terms in the row corresponding to s^3 are all zero in the above table, which has been completed by replacing 0 by ε. Now the auxiliary polynomial F(s) is found as

$$F(s) = 4s^4 + 20s^2 + 16$$

which indicates that there are two pairs of roots of equal magnitude and opposite sign on the imaginary axis. The derivative of F(s) with respect to s is

$$\frac{dF(s)}{ds} = 16s^3 + 40s$$

The terms in the s^3 row are replaced by the coefficients of the last equation, that is, 16 and 40. The array of coefficients then becomes

s^5	1	5	4	
s^4	4	20	16	
s^3	16	40	0	→ Coefficients of $dF(s)/ds$
s^2	10	16	0	
s^1	14.4	0	0	
s^0	16	0	0	

We see that there is no change in sign in the first column of the new array. Thus, the original equation has no root with a positive real part. By solving for roots of the auxiliary polynomial equation,

$$4s^4 + 20s^2 + 16 = 4(s^2 + 1)(s^2 + 4) = 0$$

we obtain $s^2 = -1, s^2 = -4$

or $s = \pm j, s = \pm j2$

These two pairs of roots are a part of the roots of the original equation. As a matter of fact, the original equation can be written in factored form as follows:

$$(s + j)(s - j)(s + j2)(s - j2)(s + 4) = 0$$

Clearly, the original equation has no root with a positive real part.

5.2.1 Relative Stability Analysis

Routh's stability criterion provides information about absolute stability. Sometimes, the engineer is interested to know information about relative stability, specially in presence of parameter variation in the process due to environmental changes or change in operating conditions, which will affect one or more of the coefficients in the characteristic equation. The relative stability may be found by shifting the imaginary axis to the left by some constant amount σ (vide Section 6.4 in Chapter 6 for frequency domain method). This can be easily done by replacing s by $s + \sigma$ in the characteristic equation and applying the Routh-Hurwitz criteria. In this case, the number of changes of sign in the first column of the array developed for the polynomial in s corresponds to the number of roots that are located to the right of the vertical line $s = -\sigma$. Thus, this test reveals the number of roots that lie to the right of the vertical line $s = -\sigma$. So, a large value of σ on the negative axis indicates that the system has high relative stability and is less prone to be unstable due to small parameter variation.

5.2.2 Control System Analysis Using Routh's Stability Criterion

By selecting only one parameter at a time in the control system as a design parameter the range of the parameter may be estimated by using Routh- Hurwitz stability criterion.

Consider the system shown in Fig. 5.2. Let us determine the range of K for stability.

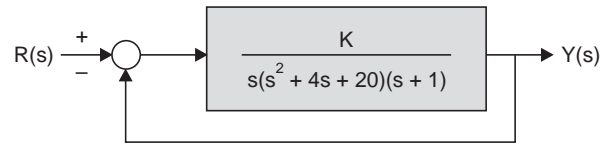



Fig. 5.2 Feedback system with variable gain K

The closed-loop transfer function is found as:

$$\frac{Y(s)}{R(s)} = \frac{K}{s(s^2 + 4s + 20)(s + 1) + K}$$

The characteristic equation is given by :

$$s^4 + 5s^3 + 24s^2 + 20s + K = 0$$

The array of coefficients becomes

s^4	1	24	K
s^3	5	20	0
s^2	20	K	0
s^1	20 – (5K/20)	0	0
s^0	K	0	0

For stability, K must be positive, and all coefficients in the first column must be positive. Therefore, for stability K must satisfy the constraint:

$$20 - (5K/20) > 0 \quad \Rightarrow \quad 0 < K < 80$$

5.3 STABILITY BY THE DIRECT METHOD OF LYAPUNOV

5.3.1 Introduction to the Direct Method of Lyapunov

The Russian mathematician A. M. Lyapunov investigated the problem of stability of dynamical systems in his doctoral thesis in 1882. He presented two methods of which the second method, or the direct method has found extensive application in the study of stability of automatic control systems.

We shall present here the second method of Lyapunov for estimating the region of stability of dynamical systems. The approach adopted here are appropriate to practical systems in contrast to purely theoretical cases, which often attempts to prove stability for any set of initial conditions, no matter how large in extent. But, in a practical situation, the concept of stability *for any initial conditions* is, very often, meaningless. If, for example, velocity is considered one of the state variables in a system, it will be absurd to allow it to assume any value, including the speed of light.

In practice, hardware limitations are more real than the theoretical limitations mentioned above. Beyond a certain velocity, the wings fall off the airplane, or, for errors larger than a few degrees, the Gyro-platform hits the gimbal, or, for flow rates beyond the pump capacity, the quality of the product is no longer acceptable. Thus the engineer is interested to use the second method of Lyapunov for investigating stability in a region of the state space that is meaningful with respect to the particular problem at hand.

The choice of the coordinate system in the generation of Lyapunov functions is an important issue. The choice of phase variables as state variables very often leads to unnecessary complications. Similarly, the choice of canonic variables, or normal coordinates, may lead to an

overly restricted region of stability after considerable effort on the part of the engineer. The physical system variables often appear to be a better choice.

5.3.2 System Representation

The application of the second method of Lyapunov for examining the stability of an autonomous, physical system assumes that dynamics of the nth-order system is represented by n simultaneous, first-order, ordinary differential equations (vide Section 3.2) of the form:

$$\begin{aligned} \dot{x}_1 &= a_{11} x_1 + a_{12} x_2 + \ldots + a_{1n} x_n \\ \dot{x}_2 &= a_{21} x_1 + a_{22} x_2 + \ldots + a_{2n} x_n \\ &\cdots \quad \cdots \quad \cdots \\ \dot{x}_n &= a_{n1} x_1 + a_{n2} x_n + \ldots + a_{nn} x_n \end{aligned} \tag{5.13}$$

In the linear system, the a_{ij}'s are constant, but more generally, the a_{ij}'s may be functions of x. In vector notation, Equation (5.13) may be written as:

$$\dot{\mathbf{x}} = A(\mathbf{x})\mathbf{x} \tag{5.14}$$

or

$$\dot{\mathbf{x}} = f(\mathbf{x}) \tag{5.15}$$

It is assumed that the variables $\mathbf{x}$ are chosen as incremental variables such that

$$f(0) = 0 \tag{5.16}$$

Quite often the system under study is represented in block diagram form. Systems with one nonlinearity, may always be represented by a nth-order differential equation of the form

$$\frac{d^n x}{dt^n} + a_n \frac{d^{n-1}x}{dt^{n-1}} + \ldots + a_2 \frac{dx}{dt} + a_1 x = 0 \tag{5.17}$$

This may be written more conveniently as: $x^{(n)} + a_n x^{(n-1)} + \ldots + a_2 \dot{x} + a_1 x = 0$ (5.18)

Equation (5.17) or (5.18) is easily reduced to n simultaneous, first-order equations by choosing the system output (or error) and its $n - 1$ derivatives as the state variables. Thus, with x_1 equal to x, and this choice of state variable, Equation (5.17) or (5.18) becomes:

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= x_3 \\ &\vdots \\ \dot{x}_n &= -a_n x_n - a_{n-1} x_{n-1} - \ldots - a_2 x_2 - a_1 x_1 \end{aligned} \tag{5.19}$$

This particular choice of state variables is referred to as the *phase variables*, a name that stems from the coordinates of the usual phase plane on which the behavior of a second-order system of the form of (5.19) is usually depicted.

For systems of order higher than two, the phase variables, are often a poor choice as system-state variables. Not only are difficulties encountered in applying the second method, but also, beyond the first derivative, these variables are not measurable. The choice of more natural system variables overcomes these disadvantages.

5.4 STABILITY BY THE DIRECT METHOD OF LYAPUNOV

Before the introduction of the Lyapunov theorems, the concepts of definiteness and closeness are considered in order to explain the precise meaning of the term stability. Only two Lyapunov stability theorems will be presented here without proof. Finally, the Cetaev instability theorem is used to point out the problems involved in initially choosing a suitable Lyapunov function.

Definiteness and Closeness of a Function

The concept of definiteness is utilized in the statement of the theorems of Lyapunov and the following definitions apply. Let the norm of x, denoted by $\| x \|$, be the Euclidean length of x, *i.e.*, $\| x \|^2 = x_1^2 + x_2^2 + ... + x_n^2$

Definition 5.1 Positive (Negative) Definite Function. *A scalar function V(x) is positive (negative) definite if, for* $\| x \| \le h$, *we have* $V(x) > 0$ [$V(x) < 0$] *for all* $x \ne 0$ *and* $V(0) = 0$.

Definition 5.2 Positive (Negative) Semi Definite Function. *A scalar function V(x) is positive (negative) semi-definite if, for* $\| x \| \le h$, *we have* $V(x) \ge 0$ [$V(x) \le 0$] *for all x and* $V(0) = 0$.

In Definitions 5.1 and 5.2, h may be arbitrarily small, in which case V would be definite in an arbitrarily small region about the origin. If h is infinite, V is definite in the entire state space.

Definition 5.3 Indefinite. *A scalar function V(x) is indefinite if it is neither Positive (Negative) Definite or Positive (Negative) Semi definite, and therefore, no matter how small the h, in the region* $|| x || \le h$, *V(x) may assume both positive and negative values.*

Example 5.3 The function $V = x_1^2 + x_2^2$ (5.20)

is positive definite if the system under consideration is second order, but it is only semi-definite if the system is third order, since, for $x_1 = x_2 = 0$, V is 0 for arbitrary x_3 . Similarly, for a third-order system, the function

$$V = x_1^2 + 2\, x_2\, x_3 + x_2^2 + x_3^2 = x_1^2 + (x_2 + x_3)^2 \tag{5.21}$$

is only positive semi definite, because, for $x_1 = 0$ and $x_2 = -\, x_3$, V is 0 at points other than the origin.

The function such as $V = x_2$ or $V = x_1 - x_3$ is obviously indefinite, no matter what the order of the system. When the function is in quadratic form, it can be expressed as

$$V = \mathbf{x}' \, P\mathbf{x} \tag{5.22}$$

Now if P is a symmetric square matrix with constant coefficients, Sylvester's theorem is used for determining the definiteness of the function

Sylvester's Theorem: *In order that the quadratic form of Equation (5.22) be positive definite, it is necessary and sufficient that the determinant of the principal minors, that is, the magnitudes*

$$| \, a_{11} \, | \, , \begin{vmatrix} a_{11} & a_{12} \\ a_{12} & a_{22} \end{vmatrix} , \text{ and } \begin{vmatrix} a_{11} & a_{12} & .. & . & a_{1n} \\ a_{12} & a_{22} & .. & . & a_{2n} \\ . & . & .. & . & . \\ . & . & .. & . & . \\ a_{1n} & a_{2n} & .. & . & a_{nn} \end{vmatrix}$$

be positive.

A scalar function W(x) is negative definite if – W(x) is positive definite.

The concept of a simple closed curve or surface is closely related to the concept of definiteness. A surface enclosing the origin is simple if it does not intersect itself, and closed if it intersects all paths that start at the origin and goes to infinity. So, a simple closed surface is topologically equivalent to the surface of an n-dimensional sphere. If V is a positive definite function, then equations $V = K_i$, $K_1 < K_2 < K_3$ ..., represent a set of nested, closed surfaces about the origin in a sufficiently small region. In order to ensure that the region extends to infinity, it is necessary to ensure that the curve V = K is closed for sufficiently large K. Such closure of

the curves V = K is assured if, in addition to positive definiteness, the function $V(x)$ approaches infinity as the norm of **x** goes to infinity; that is, if

$$\lim \| \mathbf{x} \| \to \infty, \ V(\mathbf{x}) \to \infty \tag{5.23}$$

An alternate method of testing closeness of a function $V(\mathbf{x})$ in a region is to examine the gradient of V, which is defined as a vector equal to

$$\nabla V = \begin{bmatrix} \dfrac{\delta V}{\delta x_1} \\ \dfrac{\delta V}{\delta x_2} \\ \vdots \\ \dfrac{\delta V}{\delta x_n} \end{bmatrix}$$

The function V represents a closed surface in Ω if ∇V is not zero anywhere in a region Ω containing the origin, except at the origin. The region includes the whole space, if ∇V is zero only at the origin and this is equivalent to condition (5.23). As an example of a curve that is positive definite and yet closed only for values of K less than 1, consider the function

$$V = x_2^2 + [x_1^2/(1 + x_1^2)] \tag{5.24}$$

The function $V(x)$ *does not* go to infinity as $\| x \|$ goes to infinity. Also the gradient is zero not only at $x_1 = x_2 = 0$, but also at $x_2 = 0$ and $x_1 = \infty$ .As a second example consider an integral in the Lyapunov function.

$$V = x_1^2 + \int_0^{x_1} f(x_1)\, dx_1$$

amd

$$\lim_{x_1 \to \infty} \int_0^{x_1} f(x_1)\, dx_1 = a \tag{5.25}$$

then the curve V = K is closed only for values of $K < a$.

5.4.1 Definitions of Stability

The concept of stability of a linear system with constant coefficients is basic to control engineering. Such a system is defined to be stable if and only if its output in response to every bounded input remains bounded [vide Section 5.1]. If a linear system is stable, it is stable for any input, regardless of size. This is not at all the case in nonlinear systems, as stability is a local concept and a possible function of the input. There are numerous definitions of stability. We shall define here stability only in the sense of Lyapunov and asymptotic stability, which are of interest in engineering applications.

The definitions are presented assuming that the system dynamics is available in the form of Equation (5.15).

It is further assumed that the equilibrium state being investigated is located at the origin and that $X(0) = 0$. Let S(R) be a spherical region of radius $R > 0$ around the origin, where S(R) consists of points x satisfying $|| \ x \ || < R$.

Definition 5.4 Stability in the Sense of Lyapunov. *The origin is said to be stable in the sense of Lyapunov, or simply stable, if, corresponding to each S(R), there is an S(r) such that solutions starting in S(r) do not leave S(R) as $t \to \infty$*

Definition 5.5 Asymptotic Stability. *If the origin is stable and, in addition, every solution starting in S(r) not only stays within S(R) but also approaches the origin as $t \to \infty$, then the system is called asymptotically stable.*

The definitions emphasizes the local character of types of stability for nonlinear systems, since the region S(*r*), the region of initial conditions, may be arbitrarily small. If the region S(*r*) includes the entire space, the stability defined by Definitions 5.4 and 5.5 is global.

Note that in the above the region S(R) is a function of the initial conditions or, more precisely, a function of the region of allowable initial conditions. As a consequence, a linear system with poles on the $j\omega$ axis is stable in the sense of Lyapunov. Hence, as far as automatic controls are concerned, Lyapunov stability has only academic interest. The type of stability of interest is asymptotic stability.

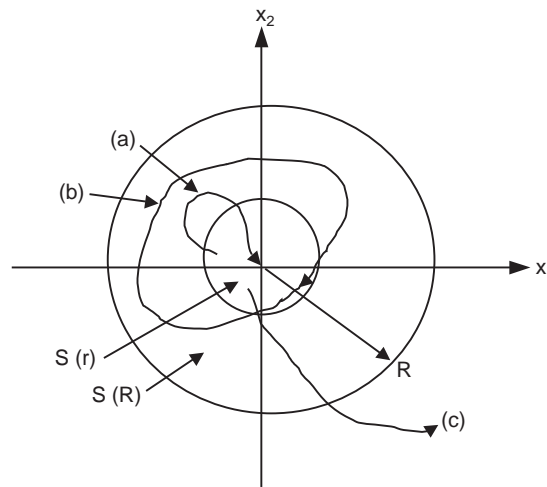



Fig. 5.3 Graphical representation of stability definitions
(*a*) Trajectory of an asymptotically stable system;
(*b*) trajectory of a system which is stable in the sense of Lyapunov;
(*c*) an unstable trajectory.

5.4.2 Lyapunov Stability Theorems

Theorem 5.1 *If there exists a function V(x), definite with respect to sign, whose total derivative with respect to time is also a function of definite sign, opposite in sense to that of V, then Equation (5.15) under assumption of Equation (5.16) is asymptotically stable.*

Theorem 5.1, the original theorem due to Lyapunov, is applicable only to an arbitrarily small region about the origin.

Modern convention assumes that V(x) is positive definite. Thus, in a geometric sense, the equations $V = K_i$, with $K_i > 0$, represent a one-parameter family of closed surfaces nested about the origin. Only local stability may be concluded, because it cannot be assumed that the surfaces remain closed in the whole space.

Theorem 5.1. is overly restrictive in two senses. First, asymptotic stability is assured in an arbitrarily small region about the origin. Second, the requirement that dV/dt be negative definite, rather than semi-definite, causes difficulties when we attempt to generate suitable Lyapunov functions for nonlinear systems. Both of these shortcomings are overcome by Theorem 5.2.

Theorem 5.2 *If there exists a real scalar function V(X), continuous with continuous first partials, such that*

(*i*) *V(X) is positive definite in a closed region Ω,*

(*ii*) *one of the surfaces V = K bounds Ω,*

(*iii*) *the gradient of V, ∇V, is not zero anywhere in Ω, except at $x = 0$,*

(*iv*) *dv/dt is negative definite or semi-definite in Ω,*

(*v*) *dV/dt is not identically zero on a solution of the system other than at $x = 0$.*

then the system described by Equation (5.15) under the assumption of Equation (5.16) is asymptotically stable in Ω.

Here conditions (*i*) and (*ii*) require that V represent a closed surface in Ω. Condition (*iii*) ensures that V has no relative maximum inside Ω, and since dV/dt is always negative, or zero, as time increases, V will decrease. In the limit, as time goes to infinity, V decreases to the origin, and with it, all the state variables simultaneously decrease to 0.

In Theorem 5.2, the requirement of Theorem I that *d*V/*dt* be negative definite is replaced by conditions (*iv*) and (*v*). These conditions require that *d*V/*dt* be only negative semi-definite, as long as it is not zero along a solution trajectory of the system. If *d*V/*dt* were zero on a solution of the system, the system would remain at that point and asymptotic stability could not be proved. If *d*V/*dt* is zero along a curve that is not a trajectory, then, if at any time a trajectory intersects such a curve, it will not remain there. Rather, the trajectory will move to a region where *d*V/*dt* is negative and V will decrease to zero.

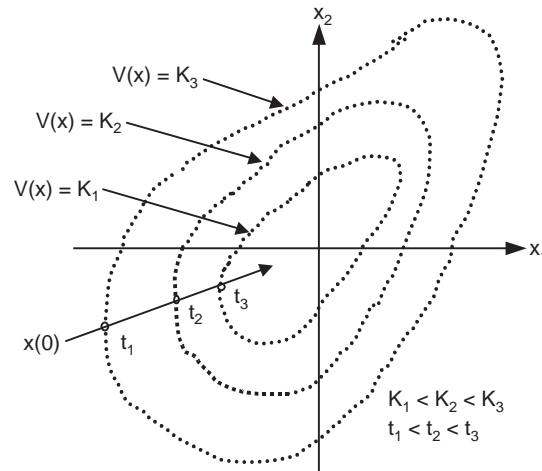



Fig. 5.4 Geometric interpretation of the stability theorem

In order to ensure that the solution to the equation *d*V/*dt* = 0 is not also a solution of system Equation (5.15), it is only necessary to substitute the solution of this equation back into Equation (5.15). In cases where the original system has more than one equilibrium point and *d*V/*dt* is semi-definite, the existence of the other equilibrium points is easily overlooked and a wrong conclusion is reached.

As pointed out earlier that, if ∇V is not equal to zero except at $x = 0$, then the region Ω extends to the entire space and global asymptotic stability may be proved.

The following instability theorem, due to Cetaev, is presented in order to highlight the basic problem that is faced in generating suitable Lyapunov functions.

Theorem 5.3 *Let S(R), be a neighborhood of the origin. Let there be a function V(x) and a region* Ω_1 *in* Ω*, with the following properties:*

(*i*) *V(x) has continuous first partials in* Ω_1,

(*ii*) *V(x) and dV/dt are positive in* Ω_1,

(*iii*) *at the boundary points of* Ω_1 *inside* Ω, *V(x) = 0,*

(*iv*) *the origin is a boundary point of* Ω_1

under these conditions, the origin is unstable.

Fig. 5.5 illustrates the case. Here V is an indefinite function and changes sign no matter how closely the origin is approached. The surface V(x) = 0 defines the boundary of Ω_1 in Ω with lines of increasing V as indicated.

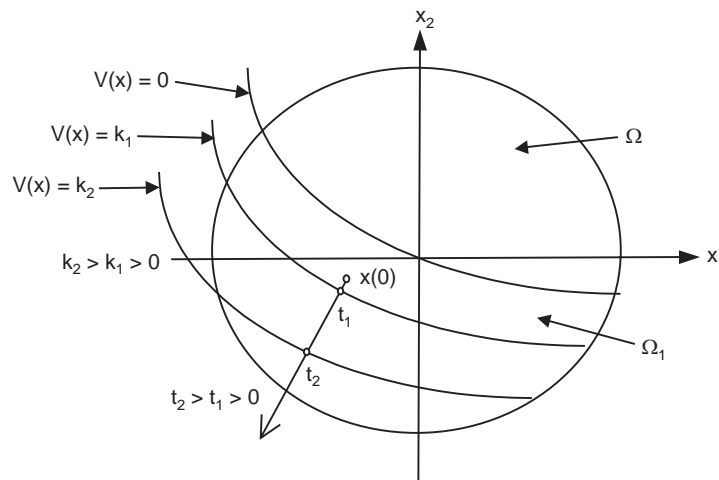



Fig. 5.5 Geometric interpretation of the Cetaev instability theorem.

Since dV/dt is positive in Ω_1 a trajectory starting from a point within Ω_1 moves in the direction of higher V, the system is unstable. If dV/dt were negative in Ω_1 and V also negative, obviously the same situation would exist. The important point here is that, if dV/dt is at least semi definite in sign, and not zero on a solution of the given system equations, then stability of the origin may be determined by an examination of V. If V is indefinite or definite and of the same sign as dV/dt, then the origin is unstable. If V is definite and of opposite sign, the origin is stable in at least an arbitrarily small region about the origin. In order to determine a bound on the size of the region, it is necessary that V and dV/dt is of opposite sign. Consequently, it is required to find the largest V curve that is closed and within which dV/dt is of opposite sign. Consequently, it is clear that an approach to the problem that guarantees some results is to ensure that dv/dt is at least semi definite in sign.

The central point of the above stability theorems is that finding a suitable Lyapunov function is only sufficient, but not necessary for ensuring stability of the system. The failure of a particular V function to prove stability in no way implies that the system in question is unstable. If dV/dt is indefinite, no conclusion can be reached. However, if dv/dt is *at least semi definite*, a conclusion can be reached.

5.5 GENERATION OF LYAPUNOV FUNCTIONS FOR AUTONOMOUS SYSTEMS

The major difficulty in applying the second method of Lyapunov is the lack of a systematic method of determining a suitable V function, especially for nonlinear systems. We shall present here a method for the generation of Lyapunov functions for linear systems only. However, we shall consider some nonlinear systems at the end of the next section and write down the representative Lyapunov functions for illustrative purposes.

5.5.1 Generation of Lyapunov Functions for Linear Systems

The generation of Lyapunov functions for linear system is almost a trivial case. For linear systems there is only one equilibrium point, the origin, and the question of stability is easily answered by the Routh-Hurwitz criteria. In view of the theorems from the previous section, as long as dV/dt is at least semi definite, stability information can be obtained from V directly. Consider the autonomous linear system.

$$\dot{x} = Ax \tag{5.26}$$

This is a special case of Equation (5.14), where $A(x)$ is not a function of x, and contains only constant elements. Let V be equal to

$$V = x' P x, \quad \text{with } p_{ij} = p_{ji} \tag{5.27}$$

The total time derivative of $V(x)$ can be written from (5.26) and (5.27) as

$$\frac{dV}{dt} = \dot{x}'Px + x'P\dot{x}$$

In view of $\dot{x}' = (Ax)' = x'A'$, we get,

$$\frac{dV}{dt} = x'A' Px + x' PAx = x' [A'P + PA]\, x = -\, x'Q\, x \tag{5.28}$$

where $$A'P + PA = -\,Q \tag{5.29}$$

Since any dV/dt is suitable, as long as it is at least semi-definite, the elements of Q may be chosen as desired, and A is also completely specified by the given system equations. Hence, only $n(n + 1)/2$ unknown elements of the P matrix need to be determined from Equation (5.29). Stability may then be concluded, if the resulting matrix P is positive definite.

An infinite number of V functions and their corresponding definite or semi definite time derivatives are all capable of proving global asymptotic stability.

Example 5.4 As an illustration of this procedure, consider the determination of the conditions for the asymptotic stability of the third-order system

$$d^3x(t)/dt^3 + a_1\, d^2x(t)/dt^2 + a_2\, dx(t)/dt + a_3\, x(t) = 0$$

Since we have no physical information about the system, let us use phase variables to write the system as three first-order equations of the form

$$\dot{x}(t) = Ax(t)$$

where $$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -a_3 & -a_2 & -a_1 \end{bmatrix}$$

Now if Q is chosen as $$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Chapter 5

then Equation (5.29) yields the following six equations for the six unknown elements of P

$$-2a_3 p_{13} = 0$$
$$p_{11} - a_2 p_{13} - a_3 p_{23} = 0$$
$$p_{12} - a_1 p_{13} - a_3 p_{33} = 0$$
$$2(p_{12} - a_2 p_{23}) = -2$$
$$p_{22} - a_1 p_{23} + p_{13} - a_2 p_{33} = 0$$
$$2(p_{23} - a_1 p_{33}) = 0$$

Solving these equations, we find P to be

$$P = \frac{1}{a_1 a_2 - a_3}\begin{bmatrix} a_1 a_3 & a_3 & 0 \\ a_3 & a_1^2 + a_2 & a_1 \\ 0 & a1 & 1 \end{bmatrix}$$

If Sylvester's theorem is applied to this matrix, the three conditions below must be satisfied if P is to be positive definite:

1. $\dfrac{a_1 a_3}{a_1 a_2 - a_3} > 0$

2. $\dfrac{a_1 a_3 (a_1^2 + a_2) - a_3^2}{(a_1 a_2 - a_3)^2} = \dfrac{a_3 [a_1^3 + (a_1 a_2 - a_3)]}{(a_1 a_2 - a_3)^2} > 0$

3. $\dfrac{a_1 a_3 (a_1^2 + a_2) - a_3^2 - a_1^3 a_3}{(a_1 a_2 - a_3)^3} = \dfrac{a_3 (a_1 a_2 - a_3)}{(a_1 a_2 - a_3)^3} = \dfrac{a_3}{(a_1 a_2 - a_3)^2} > 0$

From condition 3 it is obvious that a_3 must be positive. Using this condition in condition 1, we see that either both $a_1 a_2 - a_3$ and a_1 are positive or both are negative. However, only the first situation satisfies condition 2, and hence the conditions for asymptotic stability of the system are

$$a_1 > 0,\ a_2 > 0,\ a_3 > 0, \text{ and } a_1 a_2 - a_3 > 0$$

which is the well-known result obtained by means of the Routh-Hurwitz criterion.

In order to illustrate the arbitrariness of Q consider

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

in which case the P matrix is given by

$$P = \frac{1}{a_1 a_2 - a_3}\begin{bmatrix} a_3^2 & a_2 a_3 & 0 \\ a_2 a_3 & a_1 a_3 + a_2^2 & a_3 \\ 0 & a_3 & a_2 \end{bmatrix}$$

The conditions for P to be positive definite are then

1. $\dfrac{a_3^2}{a_1 a_2 - a_3} > 0$

2. $\dfrac{a_3^2 (a_1 a_3 + a_2^2) - a_2^2 a_3^2}{(a_1 a_2 - a_3)^2} = \dfrac{a_1 a_3^3}{(a_1 a_2 - a_3)^2} > 0$

3. $\dfrac{a_3^2(a_1a_3 + a_2^2)a_2 - a_3^4}{(a_1a_2 - a_3)^3} = \dfrac{a_3^3(a_1a_2 - a_3)}{(a_1a_2 - a_3)^3} > 0$

which can be easily shown to be equivalent to

$$a_1, a_2, a_3 > 0, \text{ and } a_1 a_2 - a_3 > 0$$

It may appear that the Routh-Hurwitz approach is more direct, but it must be remembered that the Routh-Hurwitz criterion can be applied only to a single nth-order equation or equivalently to the characteristic equation of a system. Therefore, in the general case of a system described by n first-order equations, before applying the Routh—Hurwitz criterion it is necessary to find the characteristic equation of the system or, equivalently, to put the system into phase variables. Neither one of these tasks is quite simple if the system is of high order and the A matrix has many nonzero elements. The procedure discussed above, on the other hand, does not require either of these computations, and stability can therefore be studied directly in terms of physical variables.

Example 5.5 Consider the system shown in Fig. 5.6, where the state variables have been chosen as real physical variables. The system matrix A can be written as:

$$A = \begin{bmatrix} 0 & K & 0 \\ 0 & -5 & 1 \\ -1 & 0 & -1 \end{bmatrix}$$

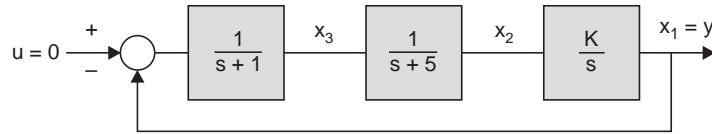



Fig. 5.6 Example of a system with physical variables as state variable

We choose a semi-definite Q matrix given by

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Then the symmetric matrix P is found from relation (5.29) as:

$$P = \frac{1}{30 - K}\begin{bmatrix} \dfrac{31K + 150}{K} & 30 + 5K & 30 - K \\ 30 + 5K & K(K + 6) & 6K \\ 30 - K & 6K & 6K \end{bmatrix}$$

In order for the quadratic function $V[x(t)] = x'(t)Px(t)$ to be positive definite, it is necessary and sufficient that

$$0 < K < 30$$

Hence the system is asymptotically stable when K is positive and less than 30.

The stability of this system could have been studied more easily using Routh-Hurwitz criterion, but it illustrates the applicability of the second method Lyapunov when the physical variables are state variable.

Chapter 5

5.6 ESTIMATION OF SETTLING TIME USING LYAPUNOV FUNCTIONS

Apart from examining stability of dynamic systems, Lyapunov functions may also be used for the estimation of settling time [19 – 20]. A class of sub optimal control systems is designed by making the approximate settling time as small as possible. We are interested here to obtain an estimate of its settling time of a linear time invariant system.

We can conclude the asymptotic stability of the system represented by

$$\dot{x} = Ax \tag{5.30}$$

by finding a positive definite Lyapunov function $V(x)$ whose total time derivative $\dot{V}(x)$ is negative definite.

Let us define a parameter T, having the dimension of time, the maximum value of the ratio of $V(x)$ to the negative of $\dot{V}(x)$ evaluated at all points in the state space, that is

$$T = \max \frac{V(x)}{-\dot{V}(x)} \tag{5.31}$$

It follows from the Equation (5.31) that

$$\dot{V}(x) \le -\frac{1}{T} V(x) \tag{5.32}$$

Dividing both sides of this equation by $V(x)$ and then integrating with respect to time from $t = 0$ to $t = t_s$, we get

$$\ln \frac{V[x(t_s)]}{V[x(0)]} \le \frac{t_s}{T} \tag{5.33}$$

or

$$V[x(t_s)] \le V[x(0)]\; e^{-t_s/T} \tag{5.34}$$

Therefore, an upper bound on the value of $V[x(t_s)]$ at any time $t_s > 0$ can be obtained by using relation (5.34) for any initial value of $V(x)$ at $t = 0$. Thus, we can find the *settling time* t_s for which the state of the system will be lying on or within the surface $V(x) = V[x(0)]\; e^{-t_s/T}$ starting from any initial state $x(0)$.

Now if $\dot{V}(x)$ is represented by a quadratic function of the form $\dot{V}(x) = -x'Qx$ where Q is a positive *definite* matrix, then $V(x)$ is also a quadratic function of the form

$$V(x) = x'Px$$

where P is the positive definite matrix obtained as the solution of $n(n + 1)/2$ unknowns from Equation (5.29). So, using (5.31) we write T as

$$T = \max_x \left[\frac{x'Px}{x'Qx}\right] \tag{5.35}$$

This division is permissible since Q is sign definite, such that $(x'Q\,x)$ is zero only at the origin $x = 0$ when $x'\,Px$ is also zero. It is convenient to evaluate the ratio in Equation (5.35) along one surface of constant $\dot{V}(x)$ in view of the fact that the relative shape and size of both $V(x)$ and $\dot{V}(x)$ remain invariant through the whole space. So we can set

$$-\dot{V}(x) = x'Qx = 1$$

The Equation (5.35) can, therefore, be recast as

$$T = \max_x\,[x'Px]$$

with the constraint that $x'Qx = 1$

The constraint can be combined by the Lagrange multiplier as

$$T = \max\,[x'Px - \lambda x'Qx]$$

where λ is chosen so that $x'Qx = 1$

Then setting the derivative of the function $[x'Px - \lambda x'Qx]$ with respect to x to zero yields

$$(P - \lambda Q)\,x = 0 \tag{5.36}$$

since both P and Q are symmetric, we have

$$Px = \lambda Qx$$

Multiplying both sides by x', we get

$$x'Px = \lambda x'Qx = \lambda$$

So, it turns out that x^TPx is maximum if λ is maximum. However, we see from Equation (5.36) that λ is an eigenvalue of $Q^{-1}P$. Hence

$$T = \text{maximum eigenvalue of } [Q^{-1}\,P] \tag{5.37}$$

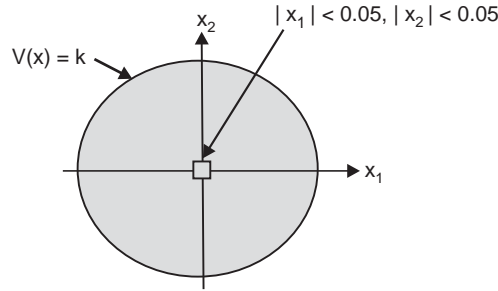



Fig. 5.7 A small zone around the equilibrium point for the estimation of settling time

Example 5.6 It is desired to find an upper bound on the time that it takes the system given below, starting from the initial condition $x'(0) = [1\ 0]$ to reach within the small zone around the equilibrium point defined by $|x_1(t_s)| < 0.05$, $|x_2(t_s)| < 0.05$

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & -2 \end{bmatrix} x$$

Let us take an positive definite Q arbitrarily as

$$Q = \begin{bmatrix} 16 & 0 \\ 0 & 16 \end{bmatrix}$$

The positive definite matrix P is found from Equation (5.29) as

$$P = \begin{bmatrix} 24 & 2 \\ 2 & 5 \end{bmatrix}$$

so that Lyapunov function is given by $V[x(t)] = 24x_1^2 + 4x_1x_2 + 5x_2^2$

Therefore, worst case value of $V\,[x(t_s)]$ can be computed as

$$x'(t_s)\,Px(t_s) = 0.0825, \text{ where } x'(t_s) = [0.05\ 0.05]$$

and $$V[x(0)] = x'(0)Px(0) = 24$$

Using Equation (5.34), the settling time t_s is given by

$$t_s \le -T\left[\ln \frac{V[x(t_s)]}{V[x(0)]}\right] \tag{5.38}$$

Now $$Q^{-1} = \begin{bmatrix} 0.0625 & 0 \\ 0 & 0.0625 \end{bmatrix} \text{ and } Q^{-1} P = \begin{bmatrix} 1.50 & 0.1250 \\ 0.1250 & 0.3125 \end{bmatrix}$$

and $$| \lambda I - Q^{-1}P | = 0 \text{ yields, } \lambda_1 = 0.2995 \text{ and } \lambda_2 = 1.5130; \text{ so } T = 1.5130 \quad (5.39)$$

Using this value of T in Equation (5.38), along with the values of $V[x(t_s)]$ and $V[x(0)]$ already computed the upper limit of settling time is found as

$$t_s \le -1.1937 \ln \frac{0.0825}{24} = 8.5834 \text{ sec} \quad (5.40)$$

So, any trajectory that starts on or within the surface $V(x) = 24$ will reach within the region defined by $|x_1(t_s)| < 0.05$, $|x_2(t_s)| < 0.05$ in less than 8.5834 sec.

This procedure provides a simple and effective method for estimating the settling time of a linear system of any order.

The approach discussed above can also be applied to nonlinear systems. But the computation is more involved, because $\dot{V}(x)$ is to be forced to be negative definite, which itself is not an easy task and second, T must be computed numerically from the defining Equation (5.31). This demonstrates of the generality of the second-method approach.

It is to be noted that the stability of a linear system can be examined by Routh-Hurwitz, Bode Diagram, Nyquist Diagram or the second method of Lyapunov. However, of all these techniques, the second method Lyapunov is applicable to the study of stability of nonlinear systems. The space constraint is preventing us to elaborate the application of Lyapunov functions for investigating stability of nonlinear systems. We only illustrate the point made above by considering the example below without mentioning the procedure for the generation of the Lyapunov function [21-23].

Example 5.7 In order to illustrate the application of Lyapunov functions for stability of nonlinear systems, consider the system in Fig. 5.8.

The equations of motion, based upon Fig. 5.8(b), may be written as:

$$\begin{aligned} \dot{x}_1 &= -3x_2 - g(x_1)\,x_1 \\ \dot{x}_2 &= -2x_2 - g(x_1)\,x_1 \end{aligned} \quad (5.41)$$

It is to be noted that the state variables here are not the phase variables.

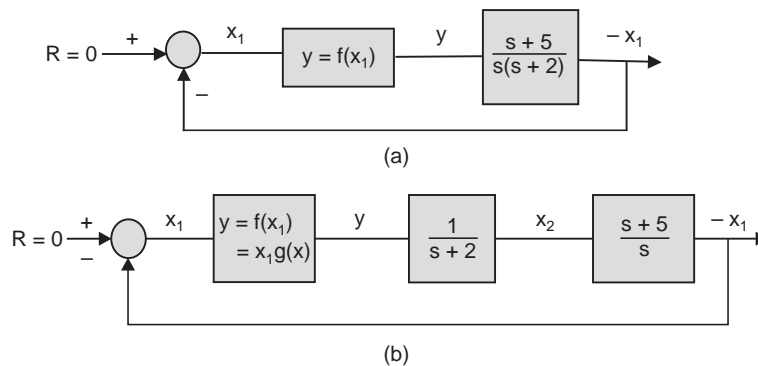



Fig. 5.8. (a) Block diagram of system for Example 5.7. (b) choice of state variables which are not phase variables. Choosing phase variables as state variables will involve derivatives of the nonlinearity in the state equation

Consider the function $$V(x) = 2 \int_0^{x_1'} g(x_1)x_1 \, dx_1 + 3x_2^2 \quad (5.42)$$

As long as $x_1 g(x_1) = f(x_1) = y$ lies in the first and third quadrant, the function V is positive definite. If the integral goes to ∞ as $x_1 \to \infty$, then V is not only positive definite, but represents a closed surface in the whole space. The total time derivative dV/dt, computed along the system trajectory (5.41) is given by :

$$\frac{dV}{dt} = 2g(x_1)x_1 \, \dot{x}_1 + 6x_2\dot{x}_2$$

Substituting the values of $\dot{x}_1$ and $\dot{x}_2$ from (5.41) and after simplification we get,

$$\frac{dV}{dt} = -\,2g^2\,(x_1)x_1^2 - 12x_2^2 \quad (5.43)$$

So, dV/dt is negative definite. Therefore, on the basis of Theorem 5.2, the system in (5.41) is globally asymptotically stable, so long the nonliearity $f(x_1)$ lies in the first and third quadrant.

Even for this simple second order system in Fig. 5.8, we can not arrive at a precise conclusion about its stability using any methods studied in earlier chapters like Routh-Hurwitz or Nyquist, since none of them is applicable here.

MATLAB SCRIPTS

```
%Script_Example5.5
%Lyapunov function for linear system with variable K
syms z A B C K Q p11 p12 p13 p22 p23 p33
A=[0 K 0;0 -5 1;-1 0 -1]; b=[0;0;1];c=[1 0 0];d=[0];A1=A′;
P = [p11 p12 p13; p12 p22 p23; p13 p23 p33]
Q=[-2 0 0;0 0 0;0 0 0]; X1=A1*P; X2=P*A; X = X2+X1;
[p11, p12, p13, p22, p23, p33] = solve('-2*p13+2=0','p11*K-5*p12-p23=0',...
'p12-p13-p33=0', 'p12*K-10*p22+K*p12=0', 'p22-6*p23+K*p13=0','2*p23-2*p33=0')
 P =[p11 p12 p13 ; p12 p22 p23; p13 p23 p33]
 D1=det(P)

% The value of X= A1*P+PA is given below. Take conj(K) = K for real K,
% then solve X= -Q

%[ -2*p13,   p11*K-5*p12-p23,   p12-p13-p33]
%[ -p23+conj(K)*p11-5*p12,   p12*K-10*p22+conj(K)*p12,   p22-6*p23+conj(K)*p13]
%[ p12-p13-p33,   p13*K-6*p23+p22,   2*p23-2*p33]
```

```
%Script_Example5.6
%Settling time by Lyapunov function.
A=[0 1;-4 -2];c=[1 0]; Q=16*eye(2,2)
P=lyap(A′,Q); % solves the special form of
%the Lyapunov matrix equation: A*P + P*A′=-Q ref Equation (5.29)
```

Chapter 5

```
% note carefully that for using lyap, use A′ not A
D1=inv(Q); D2=D1*P;D3=eig(D2); D4=max(D3)
x0=[1;0]; xts=[0.05;0.05]; V1=x0'*P*x0;
Vxts=xts'*P*xts; V=V1/Vxts; V2=log(V);
ts=D4*V2
disp ('ts is in secs')                         % Ans 8.5834 sec
b=[0;1];d=[0];sys=ss(A,b,c,d); t=0:0.05:5;
[y,t,x]=initial(sys,x0);
plot(x(:,1),x(:,2))                            % phase plane plot
%Plot(t,x)                                     % time domain plot
```

REVIEW EXERCISE

RE5.1 For each of the following characteristic equations, find the number of roots that lie to the right hand side of the s-plane by employing the Routh-Hurwitz criterion. Check the answer using MATLAB.

(*a*) $s^3 + 3s^2 - 10s + 24$ (*b*) $s^3 + 5s^2 + 5.25s + 5$

(*c*) $s^6 + 4s^5 + 5s^4 + 6s^3 + 2s + 1$ (*d*) $s^4 + 2s^3 + 2s^2 + 2s + 1$

(*e*) $s^5 - 2s^4 - 5s^3 - 12s^2 - 10s - 12$ (*f*) $s^5 + 7s^4 + 21s^3 + 33s^2 + 28s + 10$

(*g*) $s^4 + 1s^3 + 3s^2 + 3s + 2$ (*h*) $s^5 - 13s^3 + 36s$

(*i*) $s^5 + 64s$ (*j*) $s^4 + 2s^3 + s^2 - 2s - 2$

RE5.2 Find the range of stability of the closed loop systems with unity negative feedback and forward path transfer function given by :

(*a*) $\dfrac{K(s+2)}{(s^2-4)}$ (*b*) $\dfrac{K}{(s-2)(s+4)(s+25)}$ (*c*) $\dfrac{K(s+2)}{(s-2)(s-3)}$ (*d*) $\dfrac{K}{s(s+5)(s+10)}$

(*e*) $\dfrac{K}{(s+2)^3}$ (*f*) $\dfrac{K(s^2+5s+22.5)}{s(s+2)(s+3)(s+5)}$ (*g*) $\dfrac{K(s+10)}{s(s+4)}$ (*h*) $\dfrac{K(s+5)}{s(s+2)(s+10)(s+20)}$

(*i*) $\dfrac{K(s^2+8s+32)}{s(s+2)(s+3)(s+5)}$ (*j*) $\dfrac{K(s^2-2s+10)}{s(s+1)(s+2)(s+10)}$

RE5.3 Show that there are two values of K in RE5.2 (*g*) for which the damping ratio of the closed loop system is 0.7. Find the settling time (2% criteria) for each value of K when the system is subjected to a step input.

RE5.4 Find the roots of the characteristic equation of the closed loop system with unity feedback and forward path transfer function given in RE5.2 (*h*) with K = 250. Estimate the percent over shoot from the damping ratio of the dominant roots and check it from step response of the closed loop system.

RE5.5 For the RE5.2 (*i*) set K = 0.5 and find the roots of the characteristic equation using MATLAB and calculate the damping ratios of the complex roots. Estimate the per cent over shoot from the damping ratio of the dominant complex roots and check it with step response using MATLAB.

RE5.6 A system has a characteristic equation $s^3 + 5Ks^2 + (5 + K)s + 55 = 0$. Show that the system will be stable for K > 1.6533.

RE5.7 A control system has the structure shown in Fig. RE5.7. Determine the range of gain at which the system will remain stable.

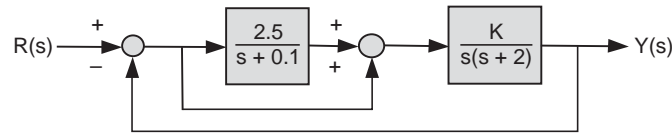



Fig. RE5.7

Ans: $0 < K < 0.84$

RE5.8 A feedback control system has a characteristic equation

$$s^3 + (10 + K)\, s^2 + 10s + (4 + 16K) = 0.$$

Find the maximum positive value K for which roots lie on the imaginary axis. Also find the frequency of oscillation.

Ans: $K = 16$, $r_{1,2} = \pm\, j\, 3.16$

RE5.9 The open-loop transfer function of a direct-drive robot arm used in manufacturing and assembly operations may be approximated by

$$G(s)\, H(s) = \frac{K(s + 8)}{(s + 2.5)(s^2 + 4s + 6)}$$

(*a*) Determine the value of gain K when the closed loop system oscillates. (*b*) Calculate the roots of the closed-loop system for the K determined in part (*a*).

Ans: $K = 59.3$, $r_1 = -6.5$, $r_{2,3} = \pm\, j8.68$

PROBLEMS

P5.1 A feedback control system has a characteristic equation

$$s^6 + 4s^5 + 6s^4 + 8s^3 + 9s^2 + 4s + 4 = 0.$$

Determine whether the system is stable, and determine the value of the roots.

P5.2 A feedback control system is shown in Fig. P5.2. The process transfer function is

$$G(s) = \frac{K(s + 20)}{s(s + 10)}$$

and the feedback transfer function is $H(s) = 1/(s + 5)$. (*a*) Determine the limiting value of gain K for a stable system. (*b*) For the gain that results in marginal stability, determine the magnitude of the imaginary roots. (*c*) Reduce the gain to 1/5 the magnitude of the marginal value and find % over shoot to step input and settling time to within 2% of steady state value.

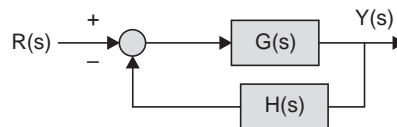



Fig. P5.2 Feedback system

Ans: (*a*) $K \le 150$, $\pm j14.1$ (*b*) with $K = 30$, over shoot = 96%, $t_s = 3.21$ sec

P5.3 Consider the population of rabbits and foxes in an island. The number of rabbits is x_1 and if left alone would grow indefinitely, until the food supply is exhausted, so that

$$\dot{x}_1 = a_1 x_1$$

However, with foxes present on the island, the growth of rabbit population is given by:

$$\dot{x}_1 = a_1 x_1 - a_2 x_2,$$

where x_2 is the number of foxes. Now, the foxes won't survive without rabbits, so the fox population is governed by:

$$\dot{x}_2 = a_3 x_1 - a_4 x_2$$

Find the constraints on a_1, a_2, a_3 and a_4 such that the system is stable and it decays to the condition $x_1(t) = x_2(t) = 0$ at $t = \infty$. Also, find the solution of $x_1(t)$ and $x_2(t)$ from $t = 0$ to 25 with $a_1 = 0.1$, $a_2 = 0.15$, $a_3 = 0.22$ and $a_4 = 0.2$ with initial values $x_o' =$ [4 1]. Repeat the solution with a_2 changed to a value = 0.23.

P5.4 An antenna position control system is shown in Fig. P5.4, where Y(s) represents the antenna position, $G_1(s)$ and G(s) represent, respectively, the amplifier and drive-motor transfer function. The wind gust on the antenna is represented by $T_d(s)$.

(a) Find the value of K_1 so as to achieve a steady state error e_{ss} less than 0.1° for $T_d(s) = 10/s$ and R(s) = 0. Also examine the effect of feedback by recalculating e_{ss} by removing the unity feedback.

(b) Determine the limiting value of gain for maintaining a stable closed loop system.

(c) It is desired to have a system settling time equal to 2 seconds. Using a shifted axis and the Routh-Hurwitz criterion, determine the value of gain that satisfies this requirement. Assume that the complex roots of the closed-loop system dominate the transient response. Does this assumption hold good here ?

$$G(s) = \frac{20}{s(s^2 + 5s + 16)}, \quad G_1(s) = \frac{K_1}{0.1s + 1}$$

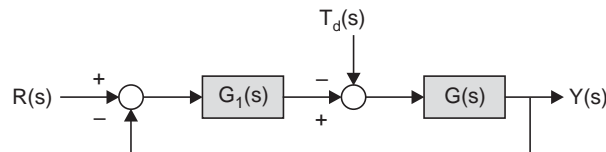



Fig. P5.4

Ans: (a) with feedback, e_{ss} = 0 and is independent of K_1; without feedback e_{ss} will be infinite (b) K_1 = 2.95 (c) t_s = 1.93 sec with K_1 = 0.78 ; No.

P5.5 Determine the relative stability of the systems with the following characteristic equations (a) by shifting the axis in the s-plane and using the Routh-Hurwitz criterion and (b) by determining the location of the complex roots in the s-plane :

(i) $s^3 + 3s^2 + 4s + 2 = 0$ **Ans:** $\sigma = -1$

(ii) $s^4 + 8s^3 + 27s^2 + 44s + 24 = 0$ **Ans:** $\sigma = -1, -2$

(iii) $s^3 + 13s^2 + 81s + 205 = 0$ **Ans:** $\sigma = -4$

P5.6 A unity-feedback control system is shown in P5.6. Determine the relative stability of the system with the following transfer functions by locating the complex roots in the s-plane.

(a) $G(s) = \dfrac{10s + 10}{s^2(s + 9)}$ (b) $G(s) = \dfrac{25}{s(s^3 + 10s^2 + 20s + 30)}$ (c) $G(s) = \dfrac{10(s + 4)(s + 5)}{s(s + 2)^2}$

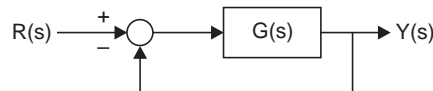



Fig. P5.6 Unity feedback system

P5.7 The block diagram of a motorized wheelchair for people paralyzed from the neck down is shown in Fig. P5.7.

The reference input is the desired velocity. Transducers are mounted in a headgear at 90° intervals so that forward, left, right, or reverse can be commanded by head movement providing output proportional to the magnitude of the movement.

(*a*) Determine the limiting gain $K = K_1K_2K_3$ for a stable system. (**Ans:** K = 12.5)

(*b*) When the gain K is set to about 40% of K found in (*a*), show that the settling time to within 2% of the final value of the system is $t_s = 5.41$ sec when subjected to a step input.

(*c*) Determine the value of gain, together with the roots of the characteristic equation, that results in a system with a settling time $\leq$ 5 sec.

(**Ans:** (*a*) K = 12.5, (*b*) K = 5.0 (*c*) with K = 4.0, $t_s = 4.78$ sec, $p_{1,2} = -0.793 \pm j2.9$, $p_3 = -6.91$)

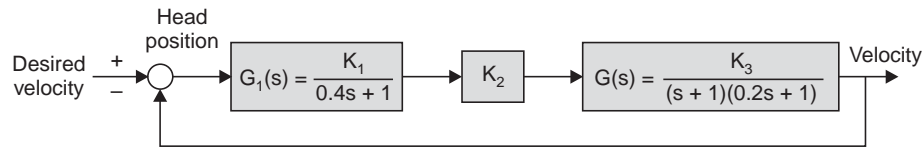



Fig. P5.7 Wheelchair control system

P5.8 A unity feedback system has a process

$$G(s) = \frac{K}{(s+4)(s-1)}.$$

Determine the range of K for which the system is (*a*) stable and (*b*) the value of K which the damping ratio $\delta = 0.6$. **Ans:** (*a*) K $\geq$ 4 (*b*) K = 10

P5.9 The block diagram of control system is shown in Fig. P5.9. (*a*) Determine the range of gain K for which the system is stable. (*b*) Also find the value of K such that the peak time for the first over shoot be less than 2 second and the percent overshoot be less than 18% .

Ans: (*a*) $0 \leq K \leq 200$ (*b*) $31.5 \leq K \leq 34.5$

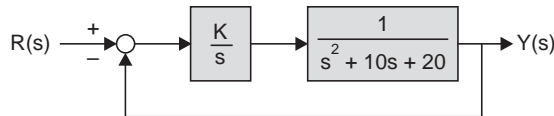



Fig. P5.9

P5.10 The technique of contactless support by magnetic suspension find wide application in light and heavy industry. A rotor is supported contactless by using magnetic suspension bearing for which the matrix differential equation is given by :

$$\dot{\mathbf{x}}(t) = A\mathbf{x}(t), \text{ where} A = \begin{bmatrix} 0 & 1 & 0 \\ -4 & -1 & 0 \\ -4 & -2 & -2 \end{bmatrix}$$

where the bearing gap *y*, its derivative *dy*/*dt* and the magnetic current I are the state vectors $x' = [x_1, x_2, x_3]$.

Examine if the system is stable by finding the roots of the characteristic equation.

Ans: The roots are -2.0, $-0.5 + j1.9365$, $-0.5 - j1.9365$ and the system is stable.

Chapter 5

P5.11 Rework Example 5.4 under the assumption that Q is

$$Q = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and show that the resulting P is

$$P = \frac{1}{a_3(a_1a_2 - a_3)} \begin{bmatrix} a_1a_2^2 - a_2a_3 + a_1^2a_3 & a_1^2a_2 & a_1a_2 - a_3 \\ a_1^2a_2 & a_1^3 + a_3 & a_1^2 \\ a_1a_2 - a_3 & a_1^2 & a_1 \end{bmatrix}$$

Show that P is positive definite for $a_1 > 0$, $a_2 > 0$, $a_3 > 0$, and $a_1\, a_2 - a_3 > 0$

P5.12 The dynamics of a second order system is given by :

$$\dot{x}(t) = Ax(t), \text{ where } A = \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix} \text{ and } x(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix}$$

Find a Lyapunov function $V(x(t))$ for the above system with $\dot{V}(x(t)) = -10x_1^2 - 10x_2^2$ and estimate the settling time, t_s, the trajectory of the system will take to reach within a close surface $V(x(t_s)) = 0.02$ around the origin, starting from $x'(0) = [1\ 1]$.

Ans: $t_s \leq 41.40$ sec

CHAPTER

6

Frequency Domain Stability Analysis and Nyquist Criterion

6.1 INTRODUCTION

We considered two time-domain approaches in Chapter 5 for investigating the stability of systems, namely Routh-Hurwitz criteria and the second method of Lyapunov. We shall now consider a frequency domain stability criterion developed by H. Nyquist in 1932 which remains a fundamental approach to the investigation of stability of linear systems [24]. The **stability criterion** presented by Nyquist is based on a theorem of Cauchy concerning the **mapping of contours** in the complex plane.

The Nyquist criterion is a semi-graphical method that determines the stability of a closed-loop system by investigating the properties of the frequency-domain plot of the loop transfer function G(s)H(s). The Nyquist method has the following features that make it suitable for the analysis and design of control systems:

It provides the same information on the absolute stability of a control system as does the Routh-Hurwitz criterion.

In addition to absolute system stability, the Nyquist criterion provides information for the relative stability or instability of a system, and provides information about improving the stability, if and when needed.

The frequency-domain plots of loop gain G($j\omega$)H($j\omega$) give information on the frequency-domain characteristics of the closed-loop system.

The stability of a non-minimum phase system including a pure time delay can be studied using the Nyquist criterion.

6.1.1 Poles and Zeros of Open Loop and Closed Loop Systems

It has been pointed out in Chapter 5 that the stability of a linear time-invariant system depends on the location of the roots of the characteristic equation with respect to the imaginary axis in the s-plane. The Nyquist criterion is used to determine the location of these roots with respect to the imaginary axis in the s-plane but it does not give the exact location of the roots of the characteristic-equation.

The closed-loop transfer function of a single-input single-output system with negative feedback is given by

$$M(s) = \frac{C(s)}{R(s)} = \frac{G(s)}{1 + G(s)\,H(s)} \tag{6.1}$$

where G(s) is the forward path and H(s) is the feedback path transfer function.

The characteristic equation is obtained by setting the denominator of the closed-loop transfer function equal to zero, that is,

$$1 + G(s)H(s) = F(s) = 0 \tag{6.2}$$

In order to include the multi-loop systems, let the characteristic equation of a closed loop system be expressed, in its most general form, as

$$F(s) = 1 + L(s) = 0 \tag{6.3}$$

where $L(s) = \dfrac{N(s)}{D(s)}$, $N(s) = K(1 + T_1 s)(1 + T_2 s) \ldots (1 + T_m s)e^{-T_d s}$

and $D(s) = s^p (1 + T_a s)(1 + T_b s) \ldots (1 + T_r s)$

Therefore, we can write the characteristic equation as

$$F(s) = 1 + L(s) = 1 + \frac{N(s)}{D(s)} = \frac{D(s) + N(s)}{D(s)} = \frac{K \prod_{i=1}^{q} (s + s_i)}{\prod_{j=1}^{n} (s + s_j)} \tag{6.4}$$

and the poles of loop gain L(s) are the poles of F(s). However, it is the zeros of F(s) that are the roots of the characteristic equation of the system which determine its response.

So we observe that

(i) *poles of loop transfer function L(s) are the poles of F(s)*

(ii) *zeros of 1 + L(s) are the poles of the closed-loop transfer function= roots of the characteristic equation*

The function D(s) reveals that there are p repeated poles of L(s) at $s = 0$ and r number of poles located at $s = -\dfrac{1}{T_a}$, $s = -\dfrac{1}{T_b}$. $s = -\dfrac{1}{T_r}$. The function L(s) also has m number of finite zeros located at $s = -\dfrac{1}{T_i}$, $i = 1, 2, \ldots\ldots m$ and one zero at $s = \infty$ corresponding to the delay term. The function F(s) has q number of finite zeros located at $s = -s_i$, $i = 1, 2, \ldots\ldots q$ and n number of finite poles located at $s = -s_j$, $j = 1, 2, \ldots\ldots n$. For a physically realizable function F(s), $n = p + r \geq q$. The function F(s) may be minimum phase or non-minimum phase types (vide Section 4.14 in Chapter 4) and the Nyquist criteria is applicable to both the types.

6.1.2 Mapping Contour and the Principle of the Argument

Before we formally state the principle of argument of complex functions, we consider a simple single valued function of s and its plot in the complex plane F(s). Let

$$F(s) = \frac{s+1}{s+5} \tag{6.5}$$

Since the argument s of the function F(s) is a complex variable given by $s = \sigma + j\omega$, the function F(s) is also a complex quantity and may be represented as $F(s) = u + jv$ that can be plotted on a complex F(s)-plane with coordinates u and v. We arbitrarily choose a close contour, designated as Γ_s, in the s-plane as shown in Fig. 6.1 (a), which does not pass through any pole and zero of F(s), that is it avoids, in the present case, the pole at $s = -5$ and zero at $s = -1$. We concentrate on the representative points $a(-2 - j2)$, $b(-2 + j0)$, $c(-2 + j2)$, $d(0 + j2)$, $e(2 + j2)$, $g(2 + j0)$, $h(2 - j2)$ and $i(0 - j2)$ on the close contour in the s-plane and calculate the corresponding points (u, v) in the F(s) plane (vide Table 6.1). The points a, b, c, d, e, g, h and i are mapped as

A , B, C, D, E, G, H and I in the F(s) plane as shown in Fig. 6.1b. By considering the intermediate points lying on the closed path *abcdeghi* in the s-plane we can obtain the closed path *ABCDEGHI*, as shown in Fig. 6.1 (b), which is designated as Γ_F.

Table 6.1

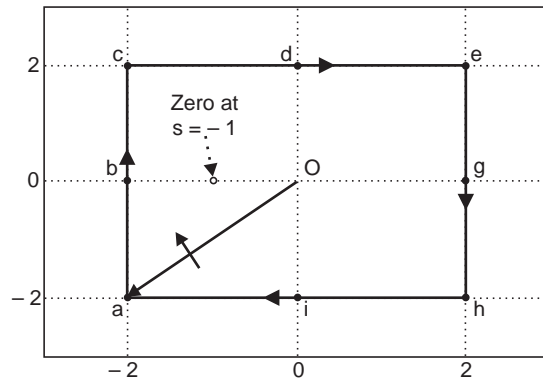
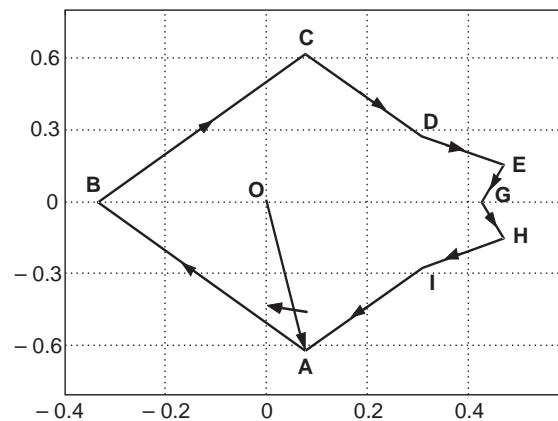
s-plane points	*F(s) plane points*
$a(-2 - j2)$	A$(0.0769 - j0.6154)$
$b(-2 + j0)$	B$(-0.3333 + j0.0)$
$c(-2 + j2)$	C$(0.0769 + j0.6154)$
$d(0 + j2)$	D$(0.3103 + j0.2759)$
$e(2 + j2)$	E$(0.4717 + j0.1509)$
$g(2 + j0)$	G$(0.4286 + j0)$
$h(2 - j2)$	H$(0.4717 - j0.1509)$
$i(0 - j2)$	I$(0.3103 - j0.2759)$

(a) s–Plane contour **abcdeghi** (Γ_s) that encloses the zero, $s = -1$, of F(s) in Equation (6.5)

(b) F(s)-plane contour ABCDEGHI (Γ_F)

Fig. 6.1 Mapping a closed contour from s-plane to F(s)-plane

We note the following facts in the mappings of Fig. 6.1 which are of interest to us.

1. The mapping of a close contour in the s-plane results in a close contour in the F(s) plane, when the chosen close contour in the s-plane does not pass through any poles and zeroes of the function F(s).
2. The origin (0,0) is inside the close contour *abcdeghi* in the s-plane. The origin (0,0) in the F(s) plane is also found to be inside the closed contour *ABCDEGHI* in the F(s) plane for the chosen closed contour abcdeghi in the s plane.
3. The zero at $s = -1$ is inside the closed contour *abcdeghi* in the s-plane whereas the pole at $s = -5$ is not.
4. As we traverse along the contour *abcdeghi* in the clockwise direction, indicated by the arrow in Fig. 6.1(a), the traversal in the contour *ABCDEGHI* in the F(s)-plane, shown by the arrow in Fig. 6.1(b), is also in the clockwise direction.

By convention, the area lying to the right as one traverses along a closed contour in the clockwise direction, is said to be *enclosed*. So, by this convention, the area inside the close contour *abcdeghi,* including the origin (0, 0) and the zero at s = – 1, in Fig. 6.1(a) is enclosed. This convention is used in control system theory and is opposite to that employed in complex variable theory but is equally applicable. The closed contour *ABCDEGHI* in Fig. 6.1(b) also encloses the origin (0, 0) and its neighbourhood area in the F(s)-plane.

Now we consider the mapping of the same function F(s) in Equation (6.5) for a *different* closed contour *abcdeghi* in the s-plane (vide Fig. 6.2(a)) such that it encloses zero at $s = -1$ as well as the pole at $s = -5$ of the function F(s). Some representative points in the s-plane along with the computed coordinates in the F(s) plane are recorded in Table 6.2. The corresponding mapping in the F(s)-plane is shown in Fig. 6.2(b), which is again a closed contour. But this time, we observe that the direction of traversal of the closed contour in the F(s)-plane is anticlockwise even though the direction of traversal in the s-plane is clockwise. This is due to the fact that we have chosen a close contour *abcdeghi* in the s-plane which encloses both the zero at $s = -1$ and the pole $s = -5$. According to the convention mentioned above, the area covered by *ABCDEGHI* in the F(s)-plane is not "enclosed", because it does not lie to the right as one moves along the closed contour *ABCDEGHI* as indicated by the arrow.

These observations will be explained in the light of the Cauchy's Theorem of conformal mapping.

The encirclement of the origin by the closed contour Γ_F in the F(s)-plane can be related to the encirclement of the origin by the closed path Γ_s in the s-plane with the help of **Cauchy's theorem** known as the **principle of the argument.** It may be stated as follows :

If a closed contour Γ_s in the s-plane encircles, while traversing in the clockwise direction, Z zeros and P poles of F(s) and does not pass through any poles or zeros of F(s), the corresponding contour Γ_F in the F(s)-plane encircles the origin of the F(s)-plane $N = Z - P$ times in the clockwise direction.

The function F(s) in relation (6.5) has a zero at $s = -1$ and a pole at $s = -5$. We find from the Fig. 6.1(a) that the path Γ_s encloses only the zero at $s = -1$. Hence, Z = 1 and P = 0 , so N = Z – P = 1 and according to the principle of argument, the contour Γ_F in the F(s)-plane should enclose the origin once, which is found to be true from Fig. 6.1(b). Similarly, the encirclement of the origin by the closed path Γ_F in Fig. 6.2(b) is zero because the path Γ_s in Fig. 6.2(a) encloses one zero (at $s = -1$) and one pole (at $s = -5$) of F(s) making Z = 1 , P = 1 and hence N = 1 – 1 = 0.

Table 6.2

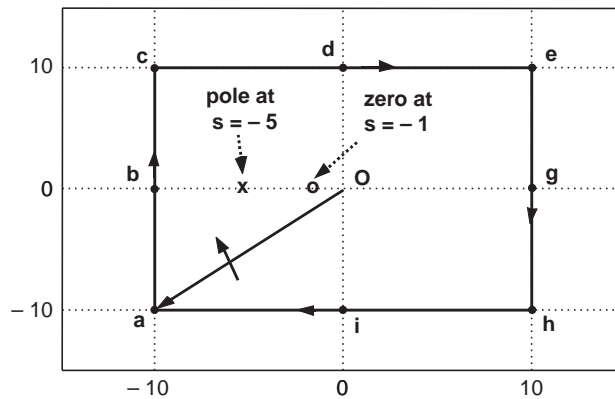
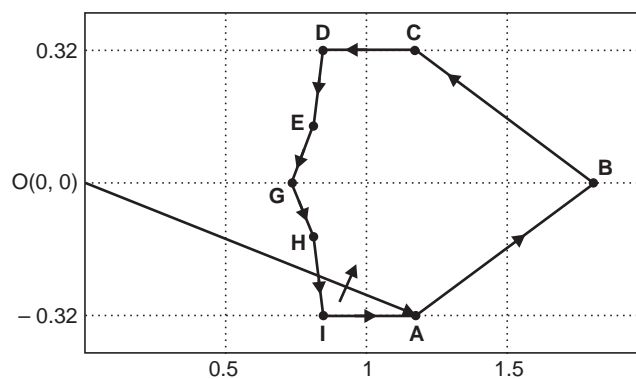
s-plane points	*F(s) plane points*
$a(-10 - j10)$	$\mathrm{A}(1.1600 - j0.3200)$
$b(-10 + j0)$	$\mathrm{B}(1.8000 + j0)$
$c(-10 + j10)$	$\mathrm{C}(1.1600 + j0.3200)$
$d(0 + j10)$	$\mathrm{D}(0.8400 + j0.3200)$
$e(10 + j10)$	$\mathrm{E}(0.8154 + j0.1231)$
$g(10 + j0)$	$\mathrm{G}(0.7333 - j0)$
$h(10 - j10)$	$\mathrm{H}(0.8154 - j0.1231)$
$i(0 - j10)$	$\mathrm{I}(0.8400 - j0.3200)$

(a) *s*-Plane contour **abcdeghi** (Γ_s) that encloses the zero, $s = -1$ and the pole, $s = -5$, of F(s) in Equation (6.5)

(b) F(s)-Plane contour ABCDEGHI (Γ_F)

Fig. 6.2 Mapping a closed contour from *s*-plane to F(*s*)-plane

The **principle of the argument** can be best understood by considering F(s) in terms of the angle due to each pole and zero as the contour Γ_s is traversed in a clockwise direction. For illustration, let us consider the function

$$F(s) = \frac{(s + z_1)}{(s + p_1)(s + p_2)(s + p_3)} \tag{6.6}$$

where z_i is a zero of F(s) and p_j , j = 1, 2, 3 are poles of F(s). Equation (6.6) may be written as

$$F(s) = | F(s) | \angle F(s) \tag{6.7}$$

where $$| F(s) | = \frac{| s + z_1 |}{| s + p_1 || s + p_2 || s + p_3 |}$$

and $\angle F(s) = (\angle s + z_1 - \angle s + p_1 - \angle s + p_2 - \angle s + p_3)$

Now for a given value of $s = s_1$ lying on an arbitrarily chosen contour Γ_s in the s-plane, the factors $(s_1 + z_1),(s_1 + p_1)$ ... in F(s) will be represented by phasors in the s-plane as shown in Fig. 6.3 such that the length of the phasor represent the magnitude contributed by the factors and the contribution to the phase angle is taken care of by the angles the phasors make with real axis measured in the anticlockwise direction. Hence, F(s) at $s = s_1$ may be written as (vide Fig. 6.3 (a))

$$F(s_1) = | F(s_1) | (\phi_{z_1} - \phi_{p_1} - \phi_{p_2} - \phi_{p_3}) \tag{6.8}$$

As one traverses in the clockwise direction (taken as positive direction here) along the contour Γ_s , the vector tips in the F(s)-plane also moves. If Γ_s does not pass through any poles and zeros of F(s), the locus Γ_F of the mapped points in the F(s)-plane , will be a closed path and if we traverse the contour Γ_s once, the angles ϕ_{p_1}, ϕ_{p_2} and ϕ_{z_1} will undergo a change of 360 degrees each whereas the net change of ϕ_{p_3}, due to the pole p_3 not encircled by Γ_s , will be zero. So for traversal of the contour Γ_s once, the total contribution to change in phase angle $\angle$F(s) is given by

$$\phi_F = \phi_{z_1} - \phi_{p_1} - \phi_{p_2} = 360° - 360° - 360° = 360° - 2(360°) = - 360°$$

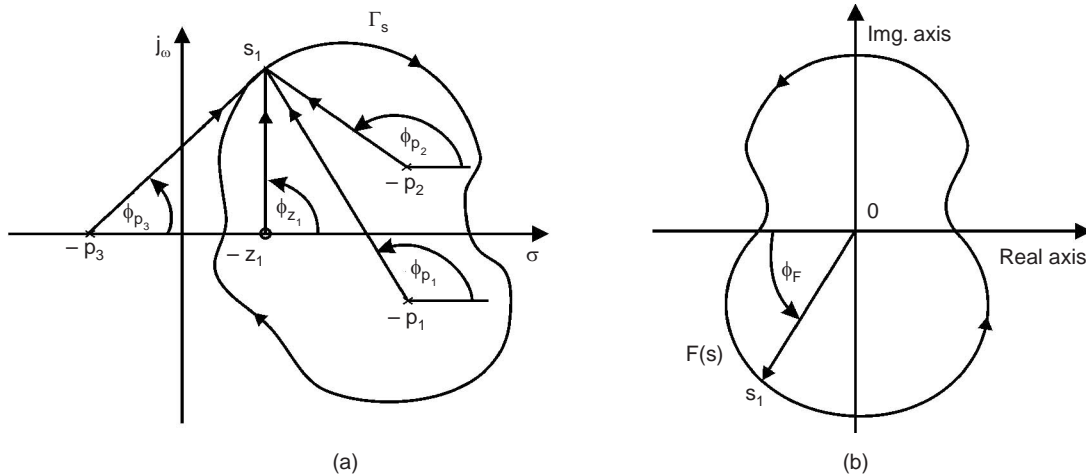



Fig. 6.3 (a) Phase angle contributed by poles and zeros (b) Phase angle of F(s)

The phase change of 360° in ϕ_F indicates that the phasor OS_1 in the F(s) plane will make one complete revolution around the origin in the F(s)-plane as the point s on Γ_s starts at s_1 and

moves along the clockwise direction and terminates at s_1. The negative sign before the phase angle, ϕ_F, indicates that rotation will be in the anticlockwise direction (*i.e.*, opposite in sense to that of Γ_s) . If the net number of revolutions traversed by this phasor is N, the net angle traversed by the phasor is $2\pi N$ radians.

In general, the above fact can be written as

$$2\pi N = 2\pi Z - 2\pi P \tag{6.9}$$

where Z is the number of zeros and P is the number of poles of F(s) encircled by the closed path Γ_s *in the s-plane while traversing in the clockwise direction and the integer N is the number of encirclement of the origin of the F(s) by the contour* Γ_F *in the clockwise direction.*

The relation (6.9) can be written as

$$N = Z - P \tag{6.10}$$

A positive value for N indicates that the encirclement of the origin in F(*s*)-plane is in the same sense (clockwise) with Γ_s and a negative value for N indicates the encirclement of the origin in the opposite sense (anticlockwise).

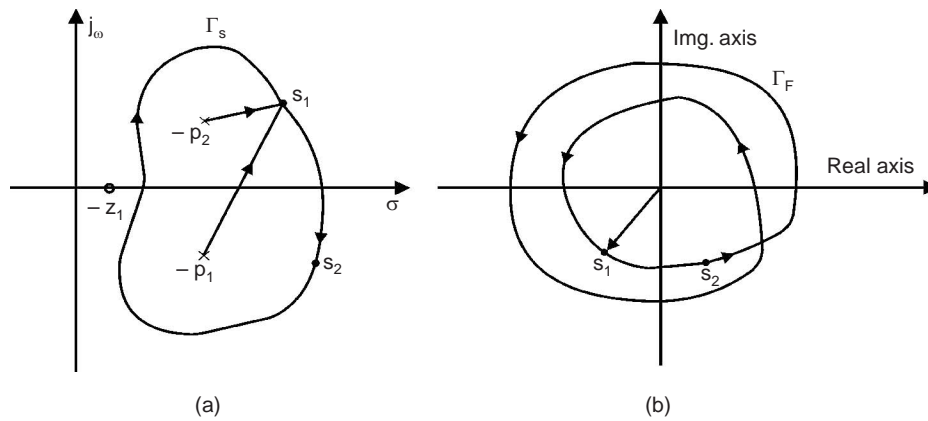



Fig. 6.4 Illustration of the principle of arguments when Γ_s encircles 2 poles

We consider the pole-zero pattern and the contour Γ_s as shown in Fig. 6.4(*a*) to illustrate the Cauchy's theorem. The contour is found to enclose two poles. Therefore we obtain in this case:

$$N = 0 - 2 = -2$$

and Γ_F completes two anticlockwise encirclements of the origin in the F(*s*)-plane, as shown in Fig. 6.4 (*b*).

Having illustrated the **Cauchy's theorem** related to the principle of arguments, we are now ready to take up the Nyquist stability criteria.

6.2 THE NYQUIST CRITERION

For stability of the closed-loop system, the zeros of the characteristic equation of F(*s*) in (6.3), must lie in the left-hand side of the imaginary axis in *s*-plane. In order to apply the Nyquist criteria, we choose a special contour Γ_s in the *s*-plane that encloses the entire right-half side of the $j\omega$ axis in the *s*-plane, and we determine whether any zeros of F(*s*) lie within Γ_F by utilizing Cauchy's theorem. This is done by plotting Γ_F in the F(*s*)-plane and counting the number of

encirclements N of the origin of F(s). Then the number of zeros of F(s) within the Γ_F contour and hence the unstable zeros of F(s) can be computed from relation (6.10) as:

$$Z = N + P \tag{6.11}$$

For a stable open loop system, the number of poles of L(s) in the right-hand s-plane is zero, hence P = 0, so the number of unstable roots of the system is equal to N, the number of encirclements of the origin of the F(s)-plane. Clearly if P = 0, we require Z = 0 for a stable system so that the contour Γ_F must not encircle the origin in the F(s)-plane. Also if P is non-zero and for a stable system Z must be zero, then we must have N = – P, or P number of *counterclockwise* encirclements of the origin in F(s)-plane.

6.2.1 The Nyquist Path

In order to apply Nyquist criteria we make a special choice of the closed contour Γ_s in the s-plane, henceforth referred to as Nyquist Path or Nyquist Contour designated as Γ_q. The Nyquist Path is shown in Fig. 6.5 which encloses the entire right-half side of the $j\omega$ axis in the s-plane while traversing in the *clockwise* direction. The contour Γ_q passes along the $j\omega$-axis from 0 to $+j\infty$, along a semicircle of radius that approaches infinity and from $-j\infty$ to 0 on the imaginary axis.

We also note from relation (6.4) that

$$L(s) = F(s) - 1 = u + jv - 1 \tag{6.12}$$

where u, v are the real and imaginary parts of F(s).

So, the real and imaginary parts of L(s) may be written as :

$$\text{Re } L(s) = -1 + u \text{ and } \text{Img } L(s) = jv.$$

So the origin of F(s) (u = 0, v = 0) is mapped to (– 1, j0) point in L(s) plane. So, if we choose the L(s)-plane, instead of F(s)-plane, for mapping the locus of the points on the Γ_q, the relation (6.11) can still be used for finding the unstable zeros of F(s) where N is the encirclement of $-1 + j0$ in the Γ_L plane. The use of the L(s) plane is convenient, since, very often, it is the loop gain G(s)H(s) of single-input-single-output system, and is available in factored form.

In view of the above discussions, we can state the Nyquist criteria as follows:

A closed-loop control system is stable if and only if the number of counterclockwise encirclements N of the point (– 1, j0) by the contour Γ_L is equal to the number of poles P of L(s) encircled by the Nyquist Path Γ_q.

For a stable open loop system, P = 0, since the number of poles encircled by Nyquist Path Γ_q is zero, the Nyquist criteria may be stated as:

A closed loop system is stable if and only if the contour Γ_L in the L(s)-plane does not encircle the (– 1, j0) point.

Let us now illustrate the Nyquist criteria by considering some typical examples.

Example 6.1 Let us consider loop gain L(s) given by

$$L(s) = G(s)H(s) = \frac{K}{(s+p_1)(s+p_2)}, \; K > 0, p_1 > 0, p_2 > 0$$

The Nyquist Path is shown in Fig. 6.5 and the contour Γ_{GH} is shown in Fig. 6.6(a) for $p_1 = 0.1$, $p_2 = 5$ and K = 10. In Table 6.3 we record some typical values of $s = j\omega$ and the corresponding values of $G(j\omega)H(j\omega) = u + jv$ used to draw the Nyquist Plot.

Table 6.3

$j\omega$	$j0$	$j0.01$	$j0.02$	$j0.03$	$j0.04$	$j0.05$	$j0.06$
u	20	19.7979	19.2151	18.3149	17.1851	15.9184	14.5979
jv	$j0$	$-j2.0198$	$-j3.9230$	$-j5.6145$	$-j7.0340$	$-j8.1592$	$-j8.9987$
$j\omega$	$j0.08$	$j0.10$	$j0.15$	$j0.30$	$j2.00$	$j50$	$j\infty$
u	12.0359	9.7961	5.8716	1.6341	-0.3010	-0.0040	-0.00
jv	$-j9.9487$	$-j10.1959$	$-j9.4069$	$-j6.0980$	$-j0.8771$	$-j0.0004$	$-j0.00$

The plot of $G(j\omega)H(j\omega)$ for $\omega = 0$ to $\omega = \infty$ is shown in continuous line. The plot of $G(j\omega)H(j\omega)$ for $\omega = -j\infty$ to $\omega = 0$ will be symmetric to the real axis u and is shown by the dashed line.

The semicircle $s = re^{j\theta}$, $\theta = \frac{\pi}{2}$ to $-\frac{\pi}{2}$ with $r \to \infty$ in the s-plane is mapped into the origin of the $G(s)H(s)$-plane.

Since the poles of $G(s)H(s)$ are at $s = -p_1 = -0.1$ and $s = -p_2 = -5$, the Nyquist Path Γ_q doesn't enclose any of them, hence $P = 0$, and from Fig. 6.6 it is observed that the encirclement, N, of the $-1 + j0$ point by Γ_{GH} is also zero. Therefore, $Z = N + P = 0 + 0 = 0$ and hence, none of the zeros of the characteristic equation $1 + G(s)H(s) = 0$ is on the right hand side of the imaginary axis in the s-plane. So, according to the Nyquist criteria, the closed-loop system is stable. We also note that the Γ_{GH} plot meets the negative real axis only at the origin as $\omega \to \infty$ and any value of $K > 0$ will not make any change in the intercept of the negative real axis of Fig. 6.6, hence the point $-1 + j0$ will never be enclosed by the Γ_{GH} for any $K > 0$, implying that the closed-loop system is always stable for $K > 0$.

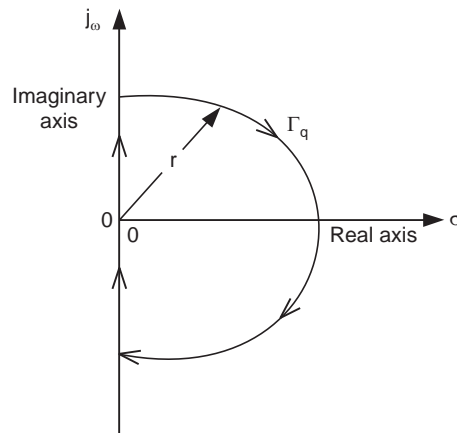



Fig. 6.5 The Nyquist path Γ_q, $\sigma \to \infty$

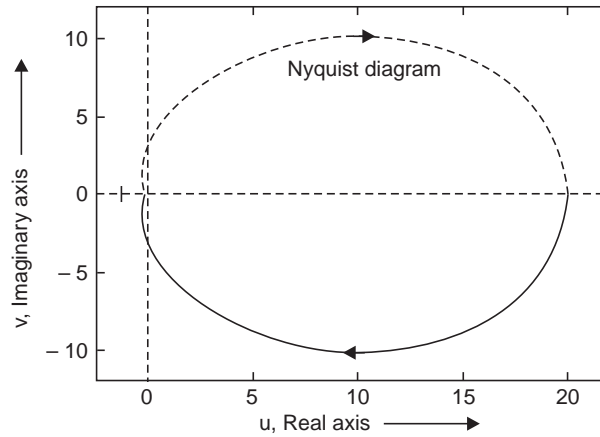



Fig. 6.6 Nyquist plot for Example 6.1; (a) plot of G(s) H(s) = $\dfrac{10}{(s+0.1)(s+5)}$

Example 6.2 Let us consider a system with $G(s)H(s) = \dfrac{50}{s(s+5)}$ having a pole at $s = 0$ and $s = -5$.

Since, the principle of arguments requires that the close contour in s should not pass through any poles of G(s)H(s), the Γ_q contour of Fig. 6.5 can't be used in this case because of the pole at the origin. The Nyquist Path is modified as shown in Fig. 6.7 (a), where a small semicircle $s = \varepsilon e^{j\theta}, \theta = -\dfrac{\pi}{2}$ to $\dfrac{\pi}{2}$ with the radius $\varepsilon \to 0$ is used to detour the origin to meet the requirements of the principle of arguments.

In order to obtain the mapping of Γ_{GH} corresponding to the modified Nyquist Path of Fig. 6.7(a), we break up the entire Nyquist Path in the following segments.

(a) *Semicircle abc (b) segment cd on imaginary axis (c) semicircle deg and (d) segment ga on imaginary axis.*

Let us now evaluate the mappings for these segments of Nyquist Path .

(a) *For the segment abc:* $s = \varepsilon e^{j\theta}, \theta = -\dfrac{\pi}{2}$ to $\dfrac{\pi}{2}$ with $\varepsilon \to 0$

$$G(s)H(s) = \lim_{\varepsilon \to 0} \frac{50}{\varepsilon e^{j\theta}(\varepsilon e^{j\theta}+5)} = \lim_{\varepsilon \to 0} \frac{50}{5\,\varepsilon\, e^{j\theta}}$$

$$= \lim_{\varepsilon \to 0} \frac{10\, e^{-j\theta}}{\varepsilon} = \infty\, e^{-j\theta}.$$

So it is a semicircle of infinite radius going from + 90° to – 90°, shown in Fig. 6.7 (b). Note that at b, $\theta = 0$ and B is mapped on the real axis at $u = +\infty$.

(b) *For the segment cd:* at d, $s = j\omega, \omega \to \infty$. $G(s)H(s) = \lim_{\omega \to \infty} \dfrac{50}{-\omega^2} = 0 \angle -180°$

So, the point D is mapped at the origin with phase of – 180°. Therefore, the mapping of the segment cd in s-plane is represented by CD in GH-plane where the point C is at $\infty \angle -90°$ and D is at $0\angle -180°$

(*c*) *For the semicircle deg:* $s = r\, e^{j\theta}$, $\theta = \frac{\pi}{2}$ to $-\frac{\pi}{2}$ with $r \to \infty$. Irrespective of the phase change, the amplitude is zero, so the entire semicircle *deg* is mapped to the origin in the GH-plane.

(*d*) *For the segment ga:* The mapping will be the mirror image about the real axis in GH-plane corresponding to that in segment *cd*.

Based on this data, the entire Γ_{GH} contour is shown in Fig. 6.7 (*b*). It is found that N = 0, so Z = N + P = 0 + 0 = 0, and hence the closed-loop system is stable. It is apparent that if the forward path gain 50 is replaced by K > 0, the – 1 + *j*0 will not be enclosed by Γ_{GH}, hence the closed-loop system is stable for any K > 0.

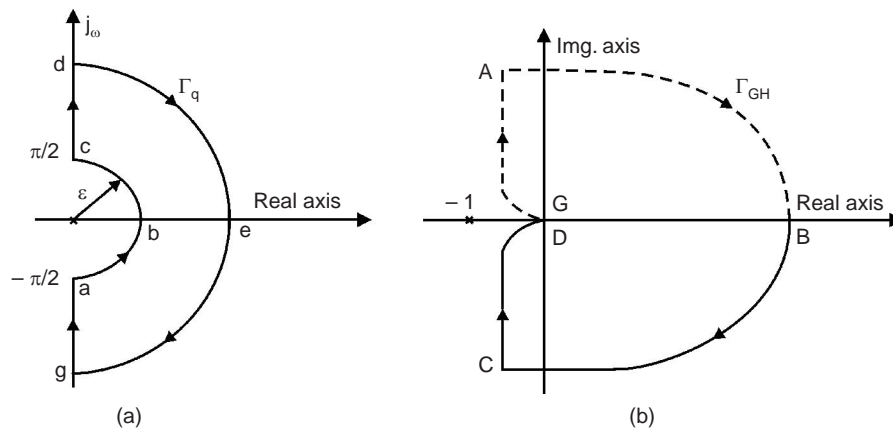



Fig. 6.7 (a) Nyquist path Γ_q with circular indentation at the origin with radius $\varepsilon \to 0$;
(b) Nyquist Plot of $G(s)H(s) = \dfrac{50}{s(s+5)}$ corresponding to Nyquist path Γ_q in (a)

Example 6.3 Let us consider the Nyquist Plot of a function given by

$$G(s)H(s) = \frac{10}{(s+0.5)(s+0.8)(s+1)(s+2.5)}$$

We choose the Nyquist Path Γ_q as shown in Fig. 6.5. The mapping of G(*s*)H(*s*) for some representative points lying on Γ_q are given in Table 6.4

Table 6.4

*j*ω	*j*0	*j*0.1000	*j*0.2000	*j*0.3000	*j*0.3700	*j*.4000
u	10	8.6625	5.4542	1.9683	– 0.0053	– 0.6823
jv	*j*0	– *j*4.3069	– *j*6.9116	– *j*7.3775	– *j*6.7688	– *j*6.3669
*j*ω	*j*0.5000	*j*0.984	*j*1	*j*10	*j*20	*j*∞
u	– 2.2126	–1.8954	– 1.8335	*j*0.0008	0.0001	0
jv	– *j*4.7710	*j*0	*j*0.0505	*j*0.0004	*j*0.0000	*j*0

The plot of the points from the table above will show the nature of the continuous line in Fig. 6.8 where the dotted line is the mirror image of the continuous line with respect to the real axis, u, in G(s)H(s)-plane and corresponding to the representative points on the $-j\omega$ axis. As before, we can show the entire semicircle of the Nyquist Path is mapped to the origin in the GH-plane. We note here that the $-1 + j0$ point is enclosed twice by Γ_{GH}, so the number of zeros of the characteristic equation of the closed-loop system lying on the right hand side of the imaginary axis is given by

$$Z = N + P = 2 + 0 = 2$$

Hence the closed loop system is unstable with two zeros of the characteristic equation lying on the right hand side of the s-plane. Recall that you have to draw a phasor from the point $(-1, j0)$ to the contour Γ_{GH} for finding N.

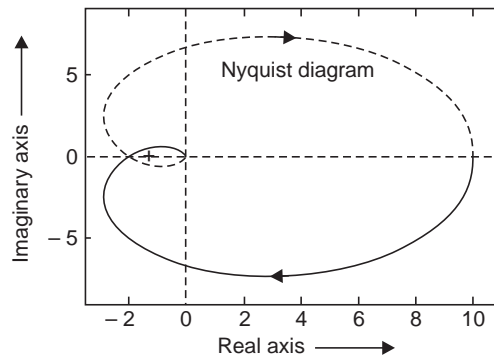



Fig. 6.8 Nyquist plot of G(jω)H(jω) for the problem in Example 6.3 when ω is varied over the entire Nyquist path

Example 6.4 Let $G(s)H(s) = \dfrac{25}{s(s+2)(s+10)}$

As in Example 6.2, the modified Nyquist Path of Fig. 6.7(a) is considered and for convenience of mapping Γ_{GH}, the entire Nyquist Path is broken up as before:

(a) Semicircle abc (b) imaginary axis cd (c) semicircle deg and (d) imaginary axis ga

Let us now evaluate the mappings for these segments of Nyquist Path .

(a) *For the segment abc* : $s = \varepsilon e^{j\theta}$, $\theta = -\dfrac{\pi}{2}$ to $\dfrac{\pi}{2}$ with $\varepsilon \to 0$

$$G(s)H(s) = \lim_{\varepsilon \to 0} \frac{25}{20\,\varepsilon e^{j\theta}} = \lim_{\varepsilon \to 0} \frac{1.25\, e^{-j\theta}}{\varepsilon} = \infty e^{-j\theta}.$$

So it is a semicircle of infinite radius going from + 90° to – 90°, shown in Fig. 6.9. Note that at b, $\theta = 0$ and B is mapped on the real axis at $u = +\infty$.

(b) *For the segment cd:* at d, $s = j\omega$, $\omega \to \infty$. $G(s)H(s) \lim_{\omega \to \infty} \dfrac{25}{-j\omega^3} = 0 \angle -270°$

So, the point D is mapped at the origin with phase of – 270° . Therefore, we find the mapping of the segment cd in s-plane is represented by CD in GH-plane where the point C is at $\infty \angle -90°$ and D is at $0 \angle -270°$.

(c) *For the semicircle deg:* $s = r\, e^{j\theta}$, $\theta = \dfrac{\pi}{2}$ to $-\dfrac{\pi}{2}$ with $r \to \infty$. Arguing as before, we see that the entire semicircle deg is mapped to the origin in the GH-plane.

(*d*) *For the segment ga:* The mapping will be the mirror image about the real axis in GH-plane corresponding to that contributed by the segment *cd*.

The intercept on the negative real axis is computed as follows :

$$G(s)H(s) = \frac{25}{s^3 + 12s^2 + 20s} \Rightarrow G(j\omega)H(j\omega) = \frac{25}{-j\omega^3 - 12\omega^2 + j20\omega}$$

After collecting terms, we get $G(j\omega)H(j\omega) = \dfrac{25}{-12\omega^2 + j(20\omega - \omega^3)}$

So the imaginary part is zero at $\omega^2 = 20$ *i.e.*, $\omega = \pm 2\sqrt{5}$ and the real part is given by

$$u = \frac{-5}{48} = -0.1042.$$

The complete plot Γ_{GH} is shown in Fig. 6.9. Hence $-1 + j0$ point will be on the left side of Γ_{GH} plot and is not enclosed by it. So N = 0 in this case and with P = 0, Z is found to be zero, which indicates that the closed loop system is stable.

In this example, a forward path gain of K > (25 × 48/5) = 240 will make the system unstable.

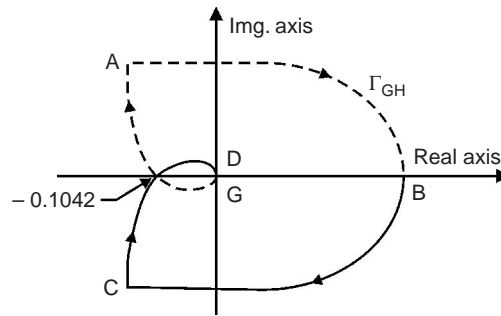



Fig. 6.9 Nyquist plot of $G(s)H(s) = \dfrac{25}{s(s+2)(s+10)}$

Example 6.5 Let $G(s)H(s) = \dfrac{K}{s^2(s+2)}$

We obtain the Nyquist Plot with K = 1 first. Following Examples 6.3 and 6.4, we obtain Γ_{GH} map by considering the following segments.

(*a*) *For the segment abc:* $s = \varepsilon e^{j\theta}, \theta = -\dfrac{\pi}{2}$ to $\dfrac{\pi}{2}$ with $\varepsilon \to 0$.

$$G(s)H(s) = \lim_{\varepsilon \to 0} \frac{1}{2\varepsilon^2 e^{j2\theta}} = \lim_{\varepsilon \to 0} \frac{0.5\, e^{-j2\theta}}{\varepsilon^2} = \infty\, e^{-j2\theta}.$$

So it is a semicircle of infinite radius going from + 180° to – 180° , shown by ABC in Fig. 6.10. We also note that at *b*, θ = 0 and B is mapped on the real axis at $u = +\infty$.

(*b*) *For the segment cd:* at d, $s = j\omega, \omega \to \infty$. $G(s)H(s) = \lim_{\omega \to \infty} \dfrac{25}{-j\omega^3} = 0 \angle -270°$

So, the point D is mapped at the origin with phase of – 270° . So the segment *cd* in s-plane is mapped to CD in GH-plane, where the point C is at $\infty \angle -180°$ and D is at $0 \angle -270°$

(*c*) *For the semicircle deg:* $s = r\, e^{j\theta}$, $\theta = \frac{\pi}{2}$ to $-\frac{\pi}{2}$ with $r \to \infty$. Irrespective of the phase change, the amplitude is zero, so the entire semicircle *deg* is mapped to the origin in the GH-plane.

(*d*) *For the segment ga:* The mapping will be the mirror image about the real axis in GH-plane corresponding to that contributed by the segment *cd.*

The intercept on the negative real axis is computed as follows :

$$G(s)H(s) = \frac{1}{s^3 + 2s^2} \quad \Rightarrow \quad G(j\omega)H(j\omega) = \frac{1}{-j\omega^3 - 2\omega^2}$$

$$\text{Rationalization yields , } G(j\omega)H(j\omega) = \frac{1}{-2\omega^2 - j\omega^3} = \frac{-2 + j\omega}{\omega^2(4 + \omega^2)}$$

So the imaginary part is zero at $\omega = \pm\infty$ and the real part is given by $u = \mp 0$

Here, we find that the point $-i + j0$ is encircled twice, N = 2. So with P = 0, Z = 2, implying the closed loop system is unstable with 2 zeros of the characteristic equation lying on the right hand side of the *s*-plane.

In this example, any forward path gain of K > 0 will make the closed loop system unstable.

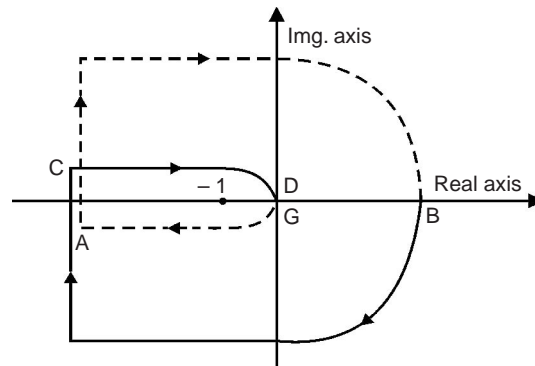



Fig. 6.10 Nyquist plot of $G(s)H(s) = \frac{K}{s^2(s+2)}$

Example 6.6 Let us now consider a G(*s*)H(*s*) with non-minimum phase given by:

$$G(s)H(s) = \frac{K(s-5)}{(s+1)(s+2)}$$

In order to draw the Nyquist diagram for this example we can use the Nyquist Path of Fig. 6.5. The contour Γ_{GH} is shown in Fig. 6.11 for K = 1. In Table 6.5 we record some typical values of $s = j\omega$ together with the corresponding values of $G(j\omega)H(j\omega) = u + jv$ which are used to draw the Nyquist Plot. The plot of $G(j\omega)H(j\omega)$ for $\omega = 0$ to $\omega = \infty$ is shown in continuous line. The plot of $G(j\omega)H(j\omega)$ for $\omega = -j\infty$ to $\omega = 0$ will be symmetric to the real axis *u* and is shown by the dashed line.

Crossing of the real and imaginary axes are found as follows:

$$G(s)H(s) = \frac{s-5}{s^2+3s+2} \Rightarrow G(j\omega)H(j\omega) = \frac{j\omega-5}{2-\omega^2+j3\omega}$$

After rationalization, we get,

$$G(j\omega)H(j\omega) = \frac{[-5(2-\omega^2)+3\omega^2]+j\omega(17-\omega^2)}{(2-\omega^2)^2+9\omega^2}$$

By setting the imaginary part equal to zero, we get ω = 0 and $\omega = \pm\sqrt{17} = \pm$ 4.123. Here the positive value of ω belongs to the positive part of the imaginary axis. The corresponding real parts of u are found to be – 2.5 and 0.3333, respectively. Similarly by setting the real part equal to zero, we get $\omega^2 = 1.25$ yielding ω = ± 1.118. Putting this value of ω, the imaginary part v is found to be $\pm$1.491. The complete plot is shown in Fig. 6.11.

Table 6.5

$j\omega$	j0	$j0.01$	$j0.05$	$j0.1$	$j0.2$	$j0.3$	$j0.5$	$j0.8$
u	– 2.500	– 2.499	– 2.487	– 2.449	– 2.304	– 2.082	– 1.506	– 0.641
jv	$j0$	$j0.043$	$j0.212$	$j0.420$	$j0.807$	$j1.138$	$j1.5771$	$j1.720$
$j\omega$	$j1.0$	$j1.5$	$j2.0$	$j4.0$	$j8$	$j20$	$j50$	$j\infty$
u	– 0.200	0.394	0.550	0.347	0.114	0.019	0.003	0.0
jv	$j1.600$	$j1.089$	$j0.650$	$j0.012$	– $j0.085$	– $j0.047$	– $j0.020$	– $j0$

From the plot of Fig. 6.11, we note that N = 1, and since P = 0 in this case Z becomes 1. So the characteristic equation has one zero in the right hand side of the s-plane and the system is unstable.

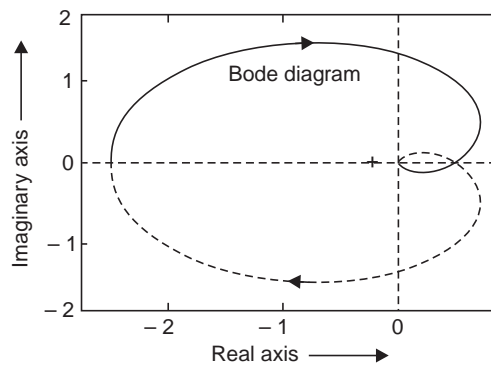



Fig. 6.11 Nyquist plot of G(s)H(s)/K = (s – 5)/[(s + 1)(s + 2)]

Example 6.7 Let us consider another non-minimum phase function given by

$$G(s)H(s) = \frac{5(s+5)}{s(s-2)}$$

The modified Nyquist Path of Fig. 6.7 (a) is considered for this example and mapping Γ_{GH} is computed as follows:

(*a*) *For the segment abc:* $s = \varepsilon\, e^{j\theta}, \theta = -\dfrac{\pi}{2}$ to $\dfrac{\pi}{2}$ with $\varepsilon \to 0$.

$$G(s)H(s) = \lim_{\varepsilon \to 0} \frac{25}{-2\varepsilon e^{j\theta}} = \lim_{\varepsilon \to 0} \frac{-12.5\, e^{-j\theta}}{\varepsilon} = -\infty\, e^{-j\theta}.$$

The negative sign before ∞ corresponds to a phase shift of – 180°. So the mapping ABC is a semicircle of infinite radius going from (– 180° + 90°) = – 90° to – 270° as shown in Fig. 6.12. Note that at *b*, θ = 0 and B is mapped on the real axis at $u = -\infty$.

(*b*) *For the segment cd:* at *d*, $s = j\omega, \omega \to \infty$. $G(s)H(s) = \lim_{\omega \to \infty} \dfrac{5j\omega}{(j\omega)^2} = 0 \angle -90°$

So, the point D is mapped at the origin with phase of – 90° . Therefore, the segment *cd* in the *s*-plane is mapped to CD of the GH-plane, the point C is at $\infty \angle -270°$ and D is at $0 \angle -90°$

(*c*) *For the semicircle deg:* Arguing as before we will find that the entire semicircle *deg* is mapped to the origin in the GH-plane irrespective of the phase change.

(*d*) For the segment *ga:* The mapping will be the mirror image about the real axis in GH-plane corresponding to that contributed by the segment *cd.*

The intercept on the negative real axis is computed as follows :

$$G(s)H(s) = \frac{5s + 25}{s^2 - 2s} \quad \Rightarrow \quad G(j\omega)H(j\omega) = \frac{25 + j5\omega}{-\omega^2 - j2\omega}$$

After the rationalization we can get

$$G(j\omega)H(j\omega) = \frac{-35\omega^2 + j5\omega(10 - \omega^2)}{\omega^4 + 4\omega^2}$$

So the imaginary part is zero at ω = 0 and $\omega^2 = 10$ yielding $\omega = \pm\sqrt{10}$

and the real part is given by $u = \dfrac{-350}{140} = -2.5$.

The complete plot Γ_{GH} is shown in Fig. 6.12.

We find that the point $-1 + j0$ is encircled once in the anticlockwise direction by Γ_{GH}. Hence , N = – 1 and with P = 1

Z = – 1 + 1 = 0 and the closed loop system is stable.

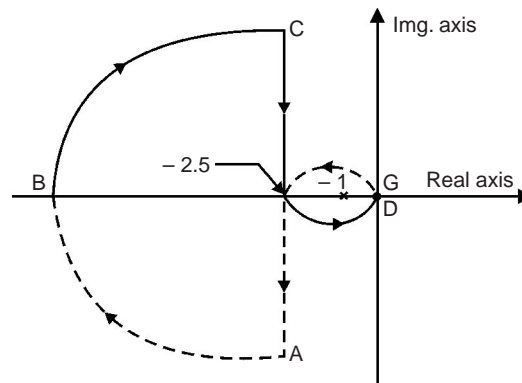



Fig. 6.12 Nyquist plot of $G(s)H(s) = \dfrac{5(s+5)}{s(s-2)}$

6.2.2 The Nyquist Plot Using a Part of Nyquist Path

The mapping of the entire Nyquist diagram Γ_L may be avoided by considering the contribution of only a part of the Nyquist Path Γ_q by using the results of Yeung and Lai [25 – 26]. They considered only the positive part of the $j\omega$ axis from $\omega = 0$ to $\omega = \infty$ on the Nyquist Path Γ_q in stating the Nyquist criteria for stability of closed loop systems. Of course, the poles of L(s) lying on the $j\omega$ axis, including those at the origin must be avoided by small circular indents for computing the Γ_L contour. The mappings from these circular indents need not be computed at all, only their count will be needed in the final analysis [vide relation (6.13) below].

Let us introduce the following variables:

P_ω = number of poles lying on the positive $j\omega$ axis including those at the origin

P = number of poles of L(s) on the right hand side of the imaginary axis in the s-plane

ϕ = The angle through which the phasor joining the $(-1 + j0)$ point to the locus Γ_L moves as ω varies from $\omega = 0$ to $\omega = \infty$ on Γ_q, excluding the contributions from the small circular indents on the $j\omega$ axis. The change in ϕ is taken to be positive when the phasor from the point $(-1 + j0)$ to Γ_L locus rotates clockwise and negative if it rotates anticlockwise.

Z = number of zeros of the characteristic equation F(s) = 1 + L(s) encircled by Γ_L, where L(s) has more poles than zeros.

From the principle of argument, Yeung and Lai have calculated the angle ϕ as:

$$\phi = (Z - 0.5\,P_\omega - P)\,180° \qquad (6.13)$$

For a stable closed loop system, Z should be zero in which case ϕ will be given by

$$\phi = -\,(0.5\,P_\omega + P)\,180° \qquad (6.14)$$

Since P_ω and P are positive integers, ϕ will be negative for a stable system. And when the system is unstable, the number of zeros of the characteristic equation encircled by the Nyquist Plot Γ_L may be found from (6.13) as

$$Z = \frac{\phi}{180°} + 0.5\,P_\omega + P \qquad (6.15)$$

Simplified Nyquist criteria: In view of the above discussions the simplified Nyquist criteria may be stated as:

A closed-loop control system is stable if and only if the change in angle ϕ defined in relation (6.13) above is negative.

For a minimum phase open loop system with P = 0, the Nyquist criteria may be stated as :

A closed-loop control system is stable if and only if the Γ_L contour, corresponding to the positive jω axis, excluding the small circular indents on Γ_q, does not enclose the (- 1 + j0) point.

Example 6.8 We consider the Example 6.4 again for which P = 0 and $P_\omega = 1$. We can follow the procedure already outlined to get the mapping CD corresponding to the part of the positive imaginary axis cd in Fig. 6.7 (a). The nature of locus CD is shown in Fig. 6.9 and since the $(-1 + j0)$ point lies to the left of the contour CD, it is not enclosed, the closed loop system is stable.

Example 6.9 Let us consider the function of Example 6.5 again for which the Nyquist path of 6.17 (*a*) is to be used.

$$G(s)H(s) = \frac{K}{s^2(s+2)}$$

For this example we note that $P_\omega = 2$ and $P = 0$. The positive part of imaginary axis on Γ_q is mapped to the contour CD in Fig. 6.10 and it is observed that the point $(-1 + j0)$ lies to the right hand side of CD , hence it is enclosed by Γ_{GH} . So the closed loop system is unstable. The number of zeros of the characteristic equation enclosed by Γ_{GH} can be found from the relation (6.13) when ϕ is measured from the Fig. 6.10 with known P_ω and P. The change in angle ϕ, the phasor from $(-1 + j0)$ makes, as the contour moves from C to D is + 180. Therefore, from (6.15) Z is computed as

$$Z = \frac{\phi}{180°} + 0.5P_\omega + P = \frac{180°}{180°} + 0.5 \times 2 + 0 = 2$$

This corroborates our earlier observations arrived at with complete Nyquist Path Γ_q in Example 6.5.

6.3 NYQUIST PLOT OF TRANSFER FUNCTION WITH TIME DELAY

So far as stability is concerned, the Nyquist plot is more flexible than the Routh-Hurwitz criteria in that it can handle transfer function with pure time delay. This is illustrated with the following Example.

Example 6.10 We consider the transfer function $G(s)H(s) = \frac{10\varepsilon^{-T_d s}}{(s+0.1)(s+5)}$, where T_d is the time delay in sec.

This type of time delay occurs in some process control system due to transportation-delay inherent in the process. For instance, in a rolling mill for producing sheets of iron with prescribed thickness, the gauge-sensors are placed downstream of the controlling cylinders that can measure the sheet thickness after a time delay. So, the error signal can be generated only after the thickness information is available from the sensors with a time delay for the controller to correct any deviation from the set value.

For this example, we choose the Nyquist Path of Fig. 6.5 and the Nyquist plot is shown in Fig. 6.13 with T_d = 1 sec. The MATLAB script file is given at the end of the Chapter.

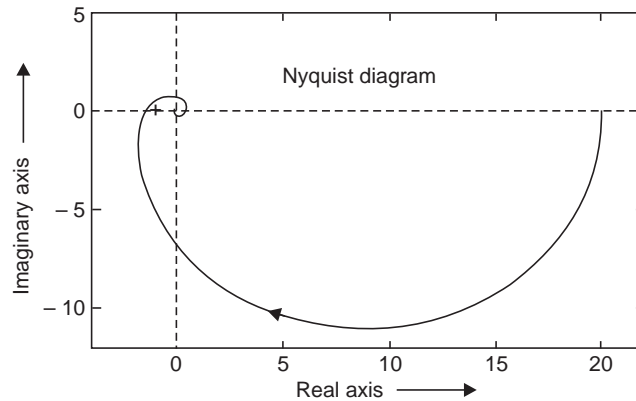



Fig. 6.13 Nyquist plot for Example 6.10 ; plot of $G(s)H(s) = \frac{10e^{-T_d s}}{(s+0.1)(s+5)}$ with T_d = 1 sec

From the Nyquist Plot in Fig. 6.13, we note that the point (– 1 + j0) is enclosed by the plot of Γ_{GH} and the closed loop system is unstable. The number of zeros of the characteristic equation encircled by the Nyquist plot is found by noting the fact that P = 0, P_ω = 0 and ϕ is found to be 360° from Fig. 6.13. Hence Z = 2.

We studied the Nyquist Plot for the same function without the pure delay term in Example 6.1 and found that the closed loop system was stable for any forward path gain K > 0. A pure delay of 1 sec makes the system unstable even with a value of K = 10.

Sometimes for the convenience of analysis and design, the pure delay function is approximated by *Pade* approximation of appropriate order, as shown below, before it is used in Nyquist Plot:

(a) First order *Pade* approximation $\varepsilon^{-T_d s} = \dfrac{1 - \dfrac{T_d s}{2}}{1 + \dfrac{T_d s}{2}}$ (6.16)

(b) Second order *Pade* approximation $\varepsilon^{-T_d s} = \dfrac{1 + \dfrac{sT_d}{2} + \dfrac{(sT_d)^2}{12}}{1 - \dfrac{sT_d}{2} + \dfrac{(sT_d)^2}{12}}$ (6.17)

The MATLAB command for Pade approximation of the delay term is : *sys* = *pade* (*g, n*); where n is the degree of the approximation polynomial and $g = \varepsilon^{-T_d s}$.

6.4 RELATIVE STABILITY: GAIN MARGIN AND PHASE MARGIN

Apart from the information of absolute stability of a system, its relative stability is equally important and Nyquist Plot comes in handy to provide a measure of relative stability. Consider the Nyquist Plot of the loop gain L($j\omega$) = G($j\omega$)H($j\omega$) for a typical third order minimum phase system with a specified forward path gain K as shown in Fig. 6.14. The point of intersection of the Nyquist Plot on the real axis is to the right hand side of the point (– 1 + j0). If the forward path gain K of the system is increased gradually, the separation of the point of intersection on the real axis from the point (– 1 + j0), will decrease gradually, until we find a critical gain K for which the

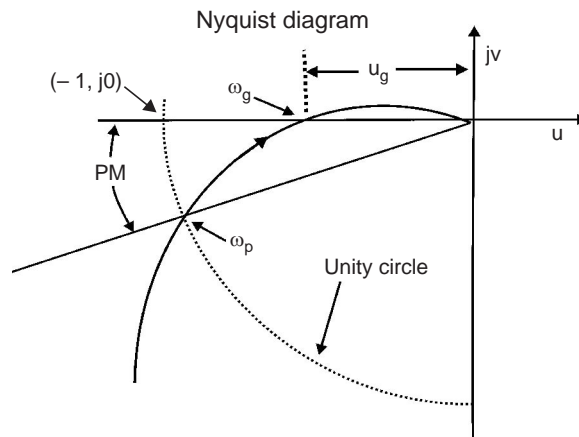



Fig. 6.14 Measure of relative stability—gain margin and phase margin

Nyquist Plot will pass through the point $(-1 + j0)$; any further increase will make the response completely unbounded. The effect of this increase in K is manifested with increased over shoot in time domain response and higher resonance peak, associated with rapid change in phase angle, in frequency response (vide Fig. 6.15). So we can define relative stability in terms of this intercept of the Nyquist Plot on the real axis.

Let the frequency for which the phase angle of the loop gain $L(j\omega)$ equals $-180°$ be designated as ω_g, and be referred to *Phase-Crossover frequency*. The length of the intercept on the real axis, measured from the origin in the L(*s*) plane, is $u_g = |L(j\omega_g)|$. The term **Gain Margin (GM)** is used to indicate a measure of relative stability of a system, which, expressed in *d*B, is defined as

$$\text{Gain Margin} = 20 \log_{10} \frac{1}{|L(\omega_g)|}\, dB = 20 \log 10 \frac{1}{|u_g|}\, dB \tag{6.18}$$

where $|u_g|$ is the intercept on the real axis by the Nyquist plot (see Fig. 6.14).

Therefore , the Gain Margin is the additional permissible gain in the loop in dB that can be increased before making the closed loop system unstable. It is clear from the expression (6.18) that if $|u_g| = 1$, Gain Margin = 0 *d*B, which means that there is no scope of any further increase of the loop gain, and when $|u_g| > 1$, Gain Margin is negative. So a negative Gain Margin indicates an unstable system. In the case of a minimum phase second order system, the value of $|u_g| = 0$, so the Gain Margin is infinite for such a system.

The parameter *Gain Margin* alone cannot provide the complete information of relative stability of a system, especially in presence of parameter variations that causes phase shift. So we introduce another term *Phase Margin* defined as follows.

Let the frequency for which the loop gain $L(j\omega)$ equals unity be designated as ω_p and referred to as *Gain-Crossover frequency* (vide Fig. 6.14). The **Phase Margin (PM),** in degrees, is defined as

$$\text{Phase Margin} = 180° - \angle L(j\omega_p) \tag{6.19}$$

Phase Margin is the angle a phasor joining the gain crossover point drawn from the origin of L(*s*) rotates to align itself with the $-180°$ line (vide Fig. 6.14). It is apparent from the definition that if the numerical value of $\angle L(j\omega_p)$ is more than 180°, the Phase Margin will be negative and the gain-crossover point will be in the third quadrant and the Nyquist Plot. Γ_L will enclose the point $(-1 + j0)$. Therefore, a negative Phase Margin corresponds to an unstable minimum phase system. It is to be noted that for conditional stable systems, there will be more than one phase cross-over frequencies and some higher order systems may have more than one gain cross-over frequencies, for which the highest gain cross-over frequency should be used for measuring phase margin.

A. Computation of Gain Margin and Phase Margin from Nyquist Plot

Example 6.11 We now consider an illustrative example for computing the Gain Margin and Phase Margin using a Nyquist diagram.

Let $$G(s)H(s) = \frac{10}{(s + 0.2)(s + 1)(s + 5)}$$

Substituting $s = j\omega$ in the above expression and rationalization, we can find the real and imaginary parts of

$$G(j\omega)H(j\omega) \text{ as : } u = \frac{10(1 - 6.2\omega^2)}{(1 - 6.2\omega^2)^2 + \omega^2 (6.2 - \omega^2)^2} \text{ and } v = \frac{-10\,\omega(6.2 - \omega^2)}{(1 - 6.2\omega^2)^2 + \omega^2 (6.2 - \omega^2)^2}$$

The phase crossover frequency ω_g is found by setting the imaginary part v equal to zero, that yields $\omega_g = \sqrt{6.2} = 2.49$ rad/sec and the intercept on the negative real axis is found to be $| u_g | = | -0.2670 |$. Therefore, the Gain Margin is computed as

$$GM = 20 \log_{10} \frac{1}{|u_g|} = 11.5\ dB$$

The Nyquist Plot will be similar to that shown in Fig. 6.14. From the plot we can find that the gain crossover frequency as $\omega_p = 1.22$ rad /sec and the corresponding phase shift is found to be

$$\angle G(j\omega_p)H(j\omega_p) = -144.9°$$

so that the Phase Margin is given by

$$PM = 180° - \angle G(j\omega_p)H(j\omega_p) = 35.1°.$$

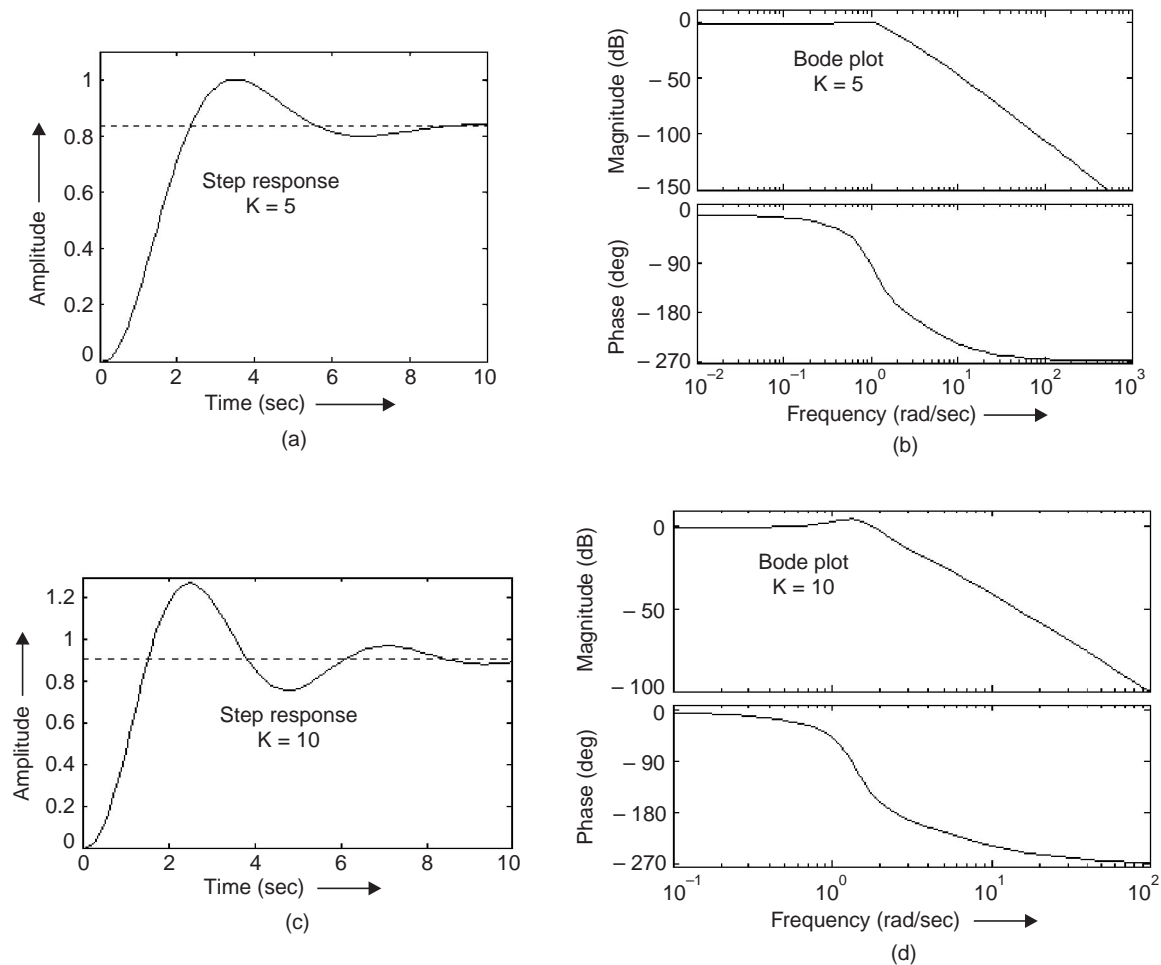



Fig. 6.18 (contd...)

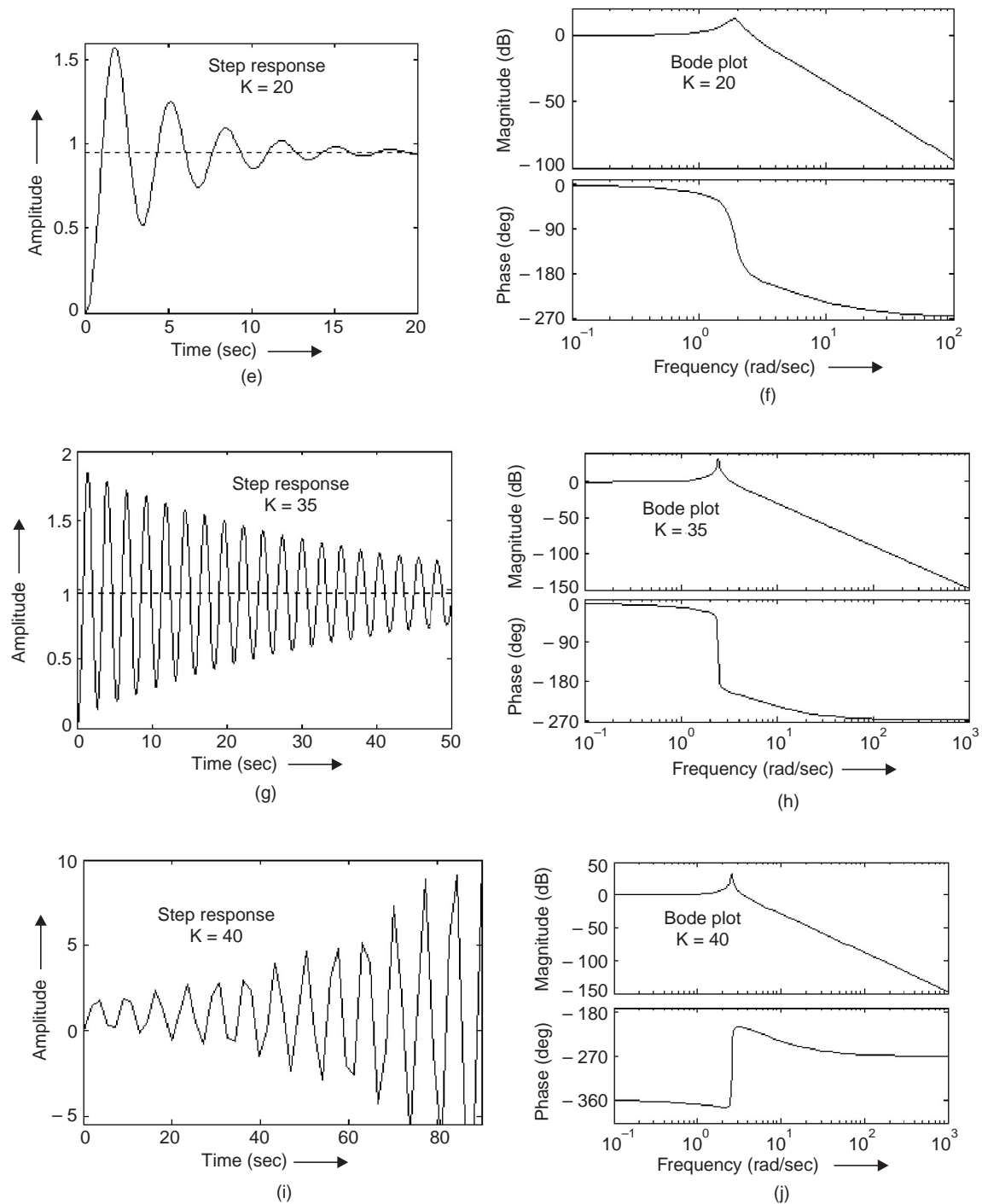



Fig. 6.15 Step and frequency response of a closed loop system with $G(s) = \dfrac{K}{(s+0.2)(s+1)(s+5)}$, and $H(s) = 1$ with different values of K

The MATLAB script for interactively observing the step response and bode plots for specified values of K in Example 6.11 is shown in Fig. 6.15 (*k*) below:

```
% Script_Fig6.15
% MATLAB script for interactively observing the step response and
 % bode plots for specified values of K in example 6.11
K=[5 10 20 35 40];
L1=length(K);
j=input( 'enter value of j in the range 1 to 5 = ');
g=zpk([],[-0.2 -1 -5],K(j));
h=zpk([],[],1);
g1=feedback(g,h);
step(g1);
a=K(j)
display ('now type >> bode(g1)')
```

Fig. 6.15 (*k*) MATLAB script for plotting Figs. 6.15 (*a*) to 6.15 (j)

B. Computation of Gain Margin and Phase Margin from Bode Plot

The parameters of relative stability—*the Gain Margin* and *Phase Margin,* as defined in relations (6.18) and (6.19), can also be found from Bode Plot. Fig. 6.16 shows the Bode Plot of $G(s)H(s) = \dfrac{10}{(s+0.2)(s+1)(s+5)}$ and the Gain Margin and Phase Margin are also indicated on the graph.

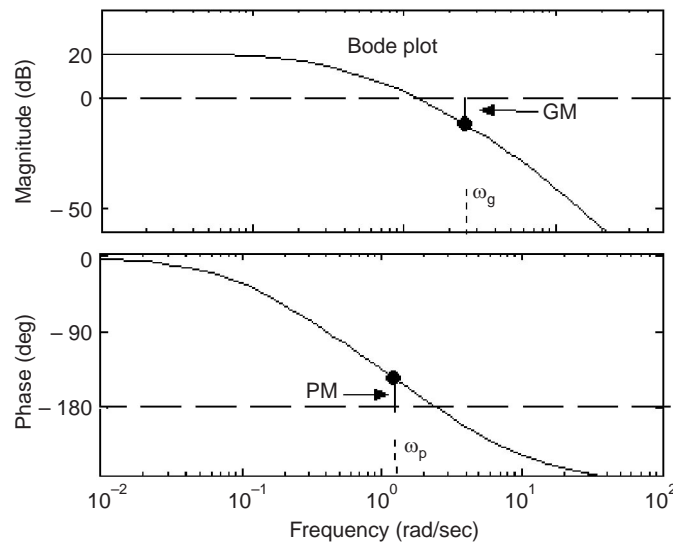



Fig. 6.16 Gain margin and phase margin of $G(s)H(s) = \dfrac{10}{(s+0.2)(s+1)(s+5)}$ using bode plot

C. The MATLAB Script for Finding GM and PM

The following MATLAB script may be used for finding the GM and PM of the transfer function in Example 6.11 without plotting the bode diagram.

```
%Script_Example6.11
gh = zpk([],[–0.2 –1 –5], 10);
[Gm, Pm, wg, wp] = margin (gh);
% The last line of the command provides GM, PM along with the
phase crossover frequency ωg (associated with GM) as well as gain
crossover frequency ωp (associated with PM) of the loop gain gh.
```

6.4.1 Analytical Expression for Phase Margin and Gain Margin of a Second Order Prototype

A second order model is widely considered as a *prototype* for the analysis and design of a control system in time domain as well as in frequency domain. This does not imply that most control systems will represent a second order dynamics. However, the dynamics of a higher order system can be approximated by a second order dynamics under favourable conditions. Quite often, there is a pair of complex conjugate poles that dominate the system response. These dominant pole pairs have smaller real parts so that they are nearer the imaginary axis in the s-plane. The contribution from other poles dies out faster than that from the dominant poles. Control engineers find this approximation of higher order dynamics by a second order system fairly accurate that simplifies the problem of analysis and design process. Of course, the approximation, must be used with care.

We are interested to derive an analytical expression for the phase margin of a second-order system and will soon discover that the phase margin is related to the damping ratio δ of an underdamped system. Consider the loop-transfer function of the system shown in Fig. 4.4 in Chapter 4 where

$$G(s)H(s) = \frac{\omega_n^2}{s(s + 2\delta\omega_n)} \tag{6.20}$$

The characteristic equation for this second-order system is given by

$$s^2 + 2\delta\omega_n s + \omega_n^{\,2} = 0$$

Therefore, the roots of the closed loop system are obtained as

$$s = -\,\delta\omega_n \pm j\omega_n \sqrt{1-\delta^2}\,,\ \delta < 1$$

The frequency response is found from Equation (6.20) by substituting $j\omega$ for s

$$G(j\omega)H(j\omega) = \frac{\omega_n^2}{j\omega\,(j\omega + 2\delta\omega_n)} \tag{6.21}$$

The frequency ω_g at which the amplitude of the loop-gain equals unity is found by solving the following equation

$$\left|\frac{\omega_n^2}{j\omega_p\,(j\omega_p + 2\delta\omega_n)}\right| = \frac{\omega_n^2}{\omega_p\,(\omega_p^2 + 4\delta^2\,\omega_n^2)^{1/2}} = 1 \tag{6.22}$$

Since the normalized frequency $\dfrac{\omega_p}{\omega_n}$ is positive, the solution, after rearranging terms, may be written as

$$\frac{\omega_p^{\,2}}{\omega_n^{\,2}} = (4\delta^4 + 1)^{1/2} - 2\delta^2 \qquad (6.23)$$

The phase margin for this system is found from (6.21) as

$$\text{PM} = \phi_m = 180° - 90° - \tan^{-1}\left(\frac{\omega_p}{2\delta\omega_n}\right)$$

$$= 90° - \tan^{-1}\left(\frac{1}{2\delta}\,[(4\delta^4 + 1)^{1/2} - 2\delta^2]^{1/2}\right) = \tan^{-1}\left(2\delta\left[\frac{1}{(4\delta^4 + 1)^{1/2} - 2\delta^2}\right]^{1/2}\right) \qquad (6.24)$$

Equation (6.24) shows the dependence of the Phase Margin on the damping ratio δ. The relation (6.24) provides a correlation between the frequency domain parameter ϕ_m with the time domain parameter δ. The variation of δ with ϕ_m is shown in Fig. 6.17. The actual variation of δ with ϕ_m in degrees may be approximated by a straight line given by

$$\phi_m = 100\,\delta \qquad (6.25)$$

which is shown in Fig. 6.17 as a dotted line.

This approximation is reasonably accurate for $\delta \le 0.7$ and is a useful index for correlating the frequency response with the transient performance of a system.

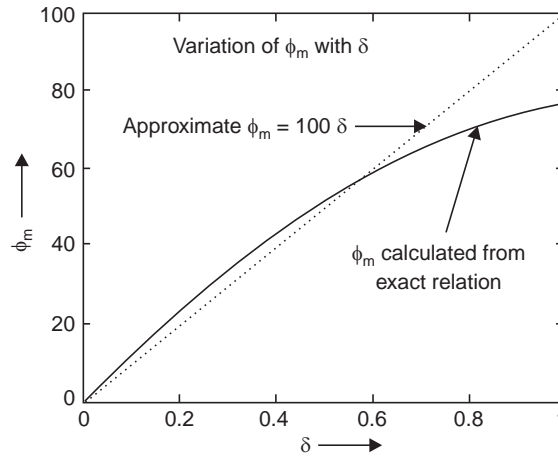



Fig. 6.17 Variation of δ with phase margin

6.5 GAIN-PHASE PLOT

Instead of plotting the amplitude and phase as a function of frequency as two separate graphs, as is done in Bode Diagram, we could have plotted the amplitude and phase on the same graph paper and the result is the Gain-Phase plot. The Gain-Phase plot has the advantage over the Bode Plot in that it may be employed to compute the frequency response of the closed loop system from the open loop response.

The graphical procedure of computing the frequency response of the closed loop system from the open loop frequency response will now be taken up.

6.5.1 Constant Amplitude (M) and Constant Phase (N) Circle

(*a*) **Constant Amplitude Loci**

The closed loop transfer function of a system, with forward path transfer function G(s) and unity feedback, is given by

$$M(s) = \frac{Y(s)}{U(s)} = \frac{G(s)}{1 + G(s)} \tag{6.26}$$

We can always represent the closed loop transfer function in the above form for a system with non-unity feedback H(s) just by taking the output of the feedback block as the system output.

Now substituting $j\omega$ for s in Equation (6.26) yields the frequency response of the closed loop system. Designating the amplitude of | M($j\omega$) | by M and representing the complex function G($j\omega$) in terms of its real and imaginary parts as G($j\omega$) = $x + jy$, the frequency response of the closed loop system may be written as

$$M = |\, M(j\omega) \,| = \left| \frac{G(j\omega)}{1 + G(j\omega)} \right| = \left| \frac{x + jy}{1 + x + jy} \right| = \frac{(x^2 + y^2)^{1/2}}{[(1 + x)^2 + y^2]^{1/2}} \tag{6.27}$$

After rearranging the terms and simplification, this can be expressed as

$$\left(x - \frac{M^2}{1 - M^2} \right)^2 + y^2 = \left(\frac{M}{1 - M^2} \right)^2, \; M \neq 1 \tag{6.28}$$

So for a given value of amplitude of the frequency response of a closed loop system, the Equation (6.28) represents the equation of a circle with radius $\left| \frac{M}{1 - M^2} \right|$ and center at (h, k)

where $h = \frac{M^2}{1 - M^2}$ and $k = 0$

So with different values of M, the Equation (6.28) represents a family of circles in the G($j\omega$) plane, referred to as **Constant-M circles**. Table 6.6, records the parameters of the circles for various values of M and Fig. 6.18 shows the loci of the circles.

Table 6.6

M–Closed loop gain with unity feedback	*Circle Centers are at (h, k)* $h = \frac{M^2}{1 - M^2}$, $k = 0$	*Circle radii* $r = \left\| \frac{M}{1 - M^2} \right\|$
M	h	r
0.2000	0.0417	0.2083
0.3000	0.0989	0.3297
0.4000	0.1905	0.4762
0.5000	0.3333	0.6667
0.6000	0.5625	0.9375
0.8000	1.7778	2.2222
1.1000	– 5.7619	5.2381
1.2500	– 2.7778	2.2222
1.4000	– 2.0417	1.4583

(contd...)

1.6000	– 1.6410	1.0256
1.8000	– 1.4464	0.8036
2.0000	– 1.3333	0.6667
3.0000	– 1.1250	– 0.3750
4.0000	– 1.0667	– 0.2667
6.0000	– 1.0286	– 0.1714

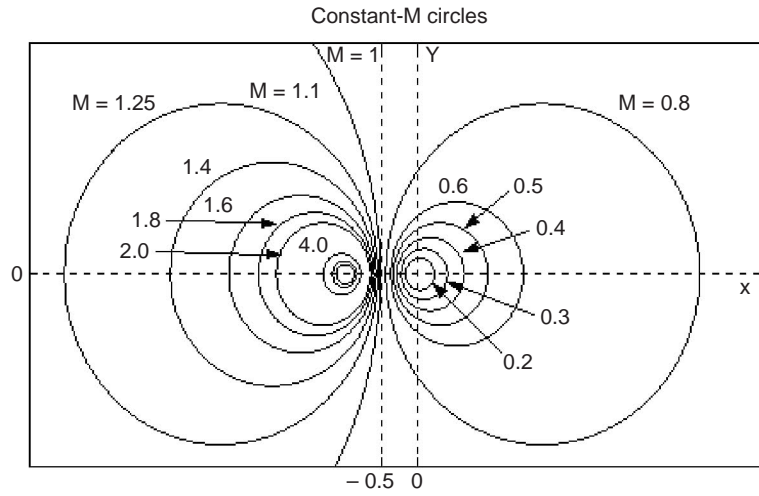



Fig. 6.18 Constant M-circle loci

(*b*) **Constant Phase-loci**

From Equation (6.27) we find the phase angle of the closed loop system is given by

$$\phi = \angle M(j\omega) = \tan^{-1}\left(\frac{y}{x}\right) - \tan^{-1}\left(\frac{y}{1+x}\right) = \tan^{-1}\left(\frac{y}{x^2+x+y^2}\right) \tag{6.29}$$

Now taking tangent of both the sides and designating tan φ = N, the above equation can be written as

$$\tan\phi = N = \frac{y}{x^2+x+y^2} \quad \Rightarrow \quad x^2 + x + y^2 - \frac{y}{N} = 0 \tag{6.30}$$

The above equation can be rearranged as

$$\left(x+\frac{1}{2}\right)^2 + \left(y - \frac{1}{2N}\right)^2 = \frac{1}{4} + \frac{1}{4N^2} \tag{6.31}$$

With constant values of N , the Equation (6.31) represents a family of circles with center at $h = -\frac{1}{2}$, $k = \frac{1}{2N}$ and radii $r = \left(\frac{N^2+1}{4N^2}\right)^{1/2}$ and are referred to as **Constant-N Circles**.

The parameters of the circles with different values of phase angles are shown in Table 6.7 and the family of circles with the corresponding values of N is shown in Fig. 6.19.

Chapter 6

Table 6.7

Phase angle of closed loop system with unity feedback ϕ	$N = \tan\phi$	*Circle Centers are at (h, k)* $h = -0.5,\ k = \dfrac{1}{2N}$	*Circle Radii,* $r = \left(\dfrac{N^2+1}{4N^2}\right)^{1/2}$
ϕ	N	k	
– 80	– 5.6713	– 0.0882	0.5077
– 60	– 1.7321	– 0.2887	0.5774
– 45	– 1.0000	– 0.5000	0.7071
– 30	– 0.5774	– 0.8660	1.0000
– 15	– 0.2679	– 1.8660	1.9319
0	0	Inf	∞
15	0.2679	1.8660	1.9319
30	0.5774	0.8660	1.0000
45	1.0000	0.5000	0.7071
60	1.7321	0.2887	0.5774
80	5.6713	0.0882	0.5077

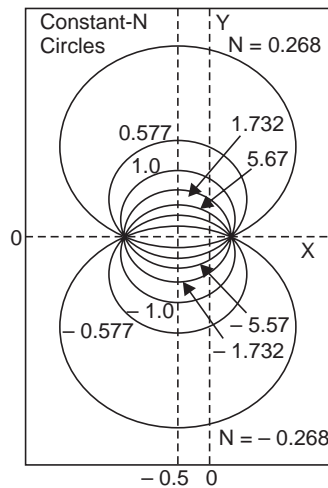



Fig. 6.19 Constant N-circles

6.6 NICHOLS PLOT

In the design stage, it is always advantageous to express the closed loop gain M in logarithmic scale as is the case for the open loop amplitude in Bode Plot. This motivated N. B. Nichols [27] to represent the Constant-M circle in *d*B and Constant-N circles in degrees in the Gain-Phase plot which is referred to as Nichols Chart (vide Fig. 6.20). The open loop frequency response is first computed with the help of Bode Plot and is then transferred to the Nichols Chart. The peak resonance M_p of the closed loop system can be found from the Nichols chart by noting the

value of M of the Constant-M circle which is touched tangentially by the open loop plot of G($j\omega$). So, the Nichols chart is a convenient tool for designing control systems with a prescribed value of peak resonance in the frequency response of the closed loop system from the Phase-Magnitude plot of the open loop system. If the forward path gain is the only design parameter to be varied, the open loop G($j\omega$) plot will be simply shifted along the vertical axis. Similarly, for incorporating only phase changes contributed by a compensating network, the G($j\omega$) plot need be shifted horizontally.

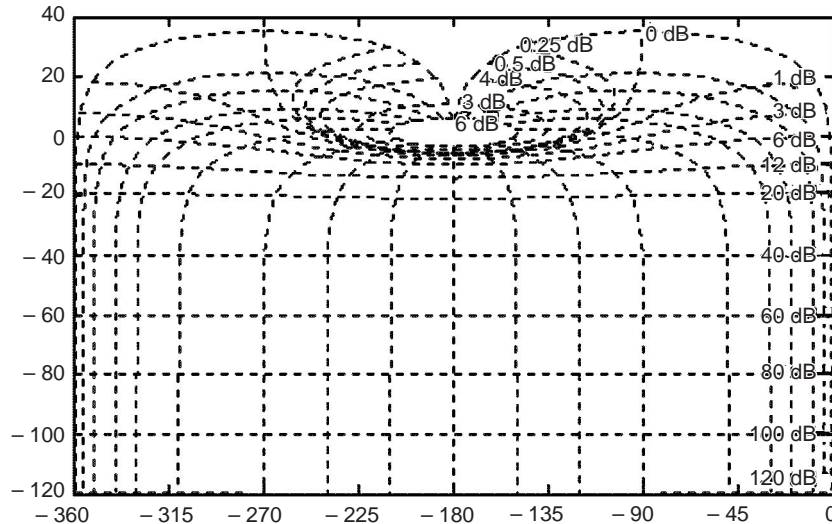



Fig. 6.20 A sample Nichols chart plotted from – 360° to 0° and – 120 dB to 40 dB

The Nichols Plot is illustrated in Fig. 6.21 for the loop gain $G(s)H(s) = \dfrac{10}{(s+0.2)(s+1)(s+5)}$

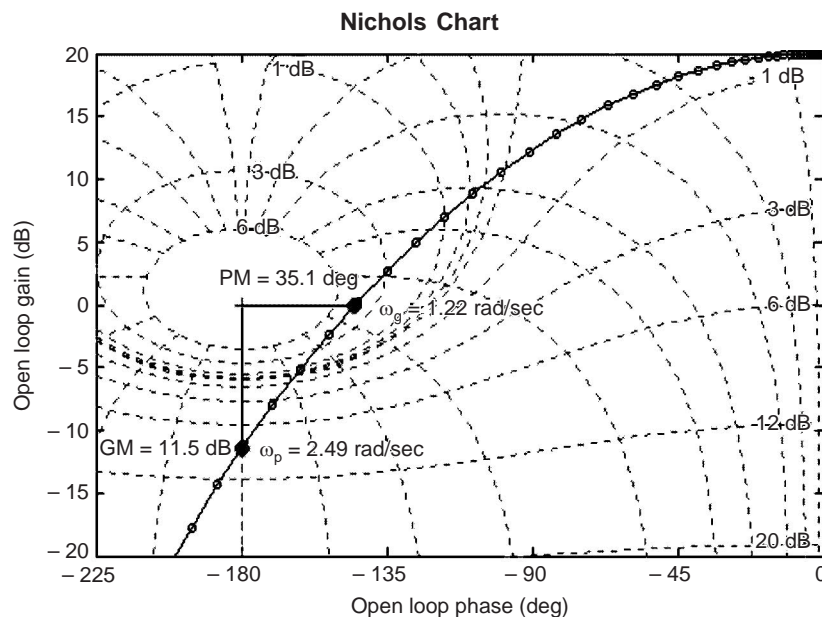



Fig. 6.21 Nichols plot for $G(s)H(s) = \dfrac{10}{(s+0.2)(s+1)(s+5)}$

6.6.1 Linear System Response Using Graphical User Interface in MATLAB

LTI Viewer is a very convenient and interactive graphical user interface (GUI) in MATLAB for analysis and comparison of time and frequency domain responses of linear system models.

Typing >> *ltiview* (*'step', sys1, sys 2*) *;* in the MATLAB command window opens an LTI Viewer containing the step response of the LTI models sys 1 and sys 2. The models sys 1, sys 2 must be created beforehand and should be present in the workspace for the above command to work. A number of plot types commonly used in linear system analysis can be used with *ltiview*. The format for the command is : *ltiview (plottype, sys 1, sys 2, ... ,sysn)* where *plottype* may specify any of the following strings , among others, or their combination to plot response in the LTI Viewer.

(*a*) 'step' Step response
(*b*) 'impulse' Impulse response
(*c*) 'bode' Bode diagram
(*d*) 'bodemag' Bode Magnitude diagram
(*e*) 'nyquist' Nyquist plot
(*f*) 'nichols' Nichols plot

Type >> *help ltiview* in the MATLAB command window to see all the options in LTI Viewer along with their command formats.

The command *ltiview({'step', 'bode'}, g1, g2)* opens an LTI Viewer showing the step as well as Bode responses of the LTI models g1 and g2.

The Nichols plot in Fig. 6.21 can be obtained with the following MATLAB script.

```
% Script Fig6.21
g=zpk([],[-0.2 –1 –5],10);
ltiview('nichols',g);
%g1=feedback(g,1);
%ltiview('bode',g1);
```

In the open LTI window, press right mouse, point to option *characteristics* and select *stability (minimum crossing)* from the drop down menu. The points corresponding to gain crossover frequency ω_p and phase crossover frequency ω_g will appear on the plot. By bringing the mouse pointer close to the indicated frequencies will show the value of GM = 11.5 *d*B, ω_g = 2.49 rad/sec and the value of PM = 35.1° and ω_p = 1.22 rad/sec along with other information. If the peak closed loop gain is to be found, select the option *grid* by pressing right mouse. After zooming around the area of interest, the peak closed loop gain is found between 3 *d*B and 6 *d*B. It is estimated as 4.5 *d*B. For more accurate value, it is convenient to use the *ltiview('bode',g1)* where g1 is the closed loop model of g with unity feedback (see script_Fig6.21 above).

MATLAB SCRIPTS

```
%Script_Fig6.6
% Plot Nyquist diagram for example 6.1
clear all; close all hidden
w = 0:0.01: 50; % range of frequency
% from 0 to 50 with increment of 0.01 rad/sec
gh = zpk([ ],[ -0.1 – 5],10);  % transfer function  G(s)H(s)
nyquist (g, w) ; % Nyquist plot of G(s)H(s)
```

```
% Script_Fig6.8
% Nyquist plot for example 6.3
clear all; close all hidden
w = 0:0.01:10;
gh = zpk([ ],[ – 0.5 – 0.8 – 1 – 2.5], 10);
nyquist (g, w);
```

```
% Script_Fig6.11
% Nyquist plot for example 6.6
clear all; close all hidden
w = 0:0.01:10;
gh = zpk([5],[- 1 - 2], 1);
nyquist (g, w);
```

```
% Script_Fig6.13
% Nyquist plot for example 6.10
clear all; close all hidden
w = 0.0:0.01:10;
gh = tf([10],[1 5.1 0.5], 'inputdelay', 1);
nyquist (gh, w);
```

```
 % Script_Fig6.17
% MATLAB script for plotting Fig. 6.17
clear all; close all hidden
d=[0:0.01:1]; L1=length(d);
 for i=1:L1; d1=d(i); d2=2*power(d1,2);
d3=power(d2,2); d4=d3+1;
den1=power(d4,0.5)-d2; den2=(1/den1);
den3=(2*d1)*power(den2,0.5);
x=atan(den3)*(180/pi);              % Equation (6.24)
p(:,i)=x;
end;
 plot(d,p,'-'); hold on
y=100*d;                            % Equation (6.25)
 plot(d,y); xlabel('\delta'),ylabel('\phim')
```

REVIEW EXERCISE

RE6.1 A system with unity feedback has forward path transfer function

$$G(s) = \frac{K(s+12)}{s(s+2)(s+8)}$$

Show that the control system is unstable when $K \geq 81.3$, and if we reduce the gain to 15, the resonant peak is 11.2 *d*B. Find the phase margin of the closed loop system with K = 15. (Ans GM=14.5 *d*B and PM = 15.9°)

RE6.2 Consider a unity feedback system with

$$G(s) = \frac{K}{s(s+1)(s+5)}$$

(*a*) Show that the gain margin is 15.6 *d*B for K = 5. (*b*) Find the value of K so as to achieve a gain margin of 21 *d*B. (*c*) Also find the phase margin of the system for the gain K found in (*b*).
Ans: (*b*) K = 2.4. (*c*) PM = 61.3°.

RE6.3 Consider a unity feedback system with

$$G(s) = \frac{40(s+2.5)}{(s+0.4)(s^2+6.25s+25)}$$

Obtain the Bode diagram using the MATLAB program or otherwise and find the gain margin and the phase margin. **Ans:** GM = ∞, PM = 50.7°

RE6.4 For the unity feedback system with

$$G(s) = \frac{15(s+2)}{s(s+0.4)(s^2+2s+25)}$$

(*a*) Find the gain margin and the phase margin.
Ans: GM = 9.06 dB, PM = 43.6°

PROBLEMS

P6.1 Sketch the polar plots of the following loop transfer functions G(*s*)H(*s*), and determine whether the system is stable by utilizing the Nyquist criterion.

(*a*) $G(s)H(s) = \frac{1}{s(s^2+s+4)}$ (*b*) $G(s)H(s) = \frac{(s+2)}{s^2\,(s+4)}$.

P6.2 Sketch the polar plot of the frequency response for the following transfer functions :

(*a*) $G(s)H(s) = \frac{K}{(1+0.4s)(1+2s)}$ (*b*) $G(s)H(s) = \frac{K(1+0.25s)}{s^2}$

(*c*) $G(s)H(s) = \frac{K(s+5)}{s^2+4s+8}$ (*d*) $G(s)H(s) = \frac{K(s+5)}{s(s+1)(s+10)}$

If the system is stable, find the maximum value for K by determining the point where the polar plot crosses the real axis.

P6.3 The pitch rate transfer function of the space shuttle is given by :

$$G(s) = \frac{0.25(s+0.04)(s^2+1600)}{(s^2+0.08s+16)(s+50)}$$

The controller is represented by $G_c(s)$ as shown in Fig. P 6.3 and the sensor is represented by a gain, H(*s*) = 0.5.

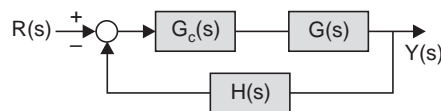



Fig. P6.3 Pitch rate control system

(*a*) Sketch the Bode diagram of the system when $G_c(s) = 2$ and find the stability margin. (*b*) Obtain the Bode plot of the system when

$$G_c(s) = K_1 + K_2/s \quad \text{and} \quad K_2/K_1 = 0.5.$$

The gain K1 should be selected so that the phase margin of the open loop system is 63°.

Ans: (*a*) ∞ (*b*) $K_i = 10$

P6.4 A unity feedback system with G(s) given by:

$$G(s) = \frac{250K}{s(s+50)(s+5)}$$

(*a*) Select a gain K so that the steady-state error for a ramp input is 10% of the magnitude of the ramp function R, where $r(t) = Rt,\ t \geq 0$. (*b*) Obtain the Bode plot of G(*s*) and determine the phase and gain margins. (*c*) Using the Nichols chart or Bode plot of the closed loop system, determine the band-width, the resonant peak M_r and the resonant frequency ω_r of the closed-loop system.

Ans: (*a*) K = 10 (*b*) PM = 31.7°, GM = 14.8 dB (*c*) BW = 10.3, rad/sec M_r = 5.28 dB, ω_r = 6.42

P6.5 A unity feedback system has a plant transfer function

$$G(s) = \frac{100}{s(s+5)}$$

(*a*) Find the Phase margin ϕ_m from bode plot. Approximating $\delta = 0.01\phi_m$, find the percent over-shoot using relation (4.14) of chapter 4. (*b*) Now obtain the step response of the closed loop system and find the percent over shoot.

Ans: (*a*) 28°, over shoot 40% (*b*) 44.4%

P6.6 The Block diagram of a closed loop system for regulating the temperature of a mold for plastic part fabrication is shown in Fig. P6.6. (*a*) Examine the stability of the system for $K_a = K = 1$ and T = 1.1 sec using Nyquist criterion.

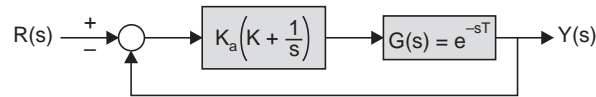



Fig. P6.6 Temperature controller

(*b*) With K = 1, find a suitable value of K_a so that the system is stable and yields a phase margin of more than 45°.

Ans: (*a*) unstable (*b*) $K_a \leq 0.8$;

Hints: Type >> *g* = *zpk* ([ ],[ ],1,'iodelay',1.1) in MATLAB command window to represent transfer function $g = e^{-1.1\,s}$

P6.7 The open-loop transfer of the remote pick-up robot arm is

$$G(s)H(s) = \frac{0.4}{s(4s+1)(s+2.5)}$$

(*a*) Plot the Nichols chart and show that the gain margin is approximately 24.7 *d*B. (*b*) Determine the phase margin and the closed-loop bandwidth.

Ans: (*b*) Phase Margin = 57.6° ; closed loop bandwidth = 0.23 rad/sec

P6.8 A closed-loop system has the loop transfer function

$$G(s)H(s) = \frac{Ke^{-Ts}}{s}.$$

(*a*) Obtain the Nyquist plot of G(*s*)H(*s*) with T = 0.1 sec and K = 1 and find the phase margin. (*b*) Also find K such that phase margin is 45°. (*c*) With K set as in (*b*) obtain a plot of the phase margin by varying T from 0.025 to 0.2 in steps of 0.025 sec.

Ans: (*a*) 84.27° (*b*) K = 7.8.

CHAPTER

7

Root Locus Technique

7.1 CORRELATION OF SYSTEM–ROOTS WITH TRANSIENT RESPONSE

We have observed in Section 4.5.1 in Chapter 4 that the time domain response of a closed loop system depends on the location of the roots of the characteristic equation. The dominant complex conjugate pair of closed loop poles is associated with overshoot and undershoots in transient response to a step input. When the roots lie on the negative real axis, the transient response does not exhibit any overshoot or undershoot but the response is somewhat sluggish. Also, if a pair of roots lies on the imaginary axis, the transient response exhibits sustained oscillations. If more than one pair of roots lie on the imaginary axis or any root has positive real part, the transient response is unbounded and the system is unstable. This correlation is well understood for a system and this knowledge is used in realizing a desired transient response by suitably placing the dominant pole-pairs in the s-domain.

Because of the close correlation of transient responses with the roots of the characteristic equation of a system, we are interested to study the effect of varying some design parameters on the variation of the roots. This is precisely the objective of our next section to study the root locus technique, first introduced by Evans in 1948 [28].

7.2 THE ROOT LOCUS DIAGRAM—A TIME DOMAIN DESIGN TOOL

In view of the discussions in the previous section, the objective of designing controllers for a system will be to place the closed loop poles in suitable locations to get a desired output response in a given application. The desired response, in turn, depends on the application itself. For example, if the roots of a second order position control system lies on the negative real axis in the s-plane, the step response is over damped with long rise time, and if the roots are complex conjugate with negative real parts, the step response shows overshoot and undershoot but the rise time is smaller. So if a feedback system is used to control the position of an antiaircraft gun, we will prefer the step response of the second kind where the rise time is smaller, even if it may be associated with large overshoot and undershoot. This is the requirement of the very nature of application, where the output-the gun position, should be equal to the command input- corresponding to the position of the enemy aircraft (in vertical plane, say), in minimum possible time, even if the gun turret moves away from the target due to oscillatory nature of its response. This is not a problem since the gunners never practice economy in ammunition and they will not allow the enemy aircraft time to release its ammunition. The response of the control system for positioning the Hull of a ship, on the other hand, should not exhibit any overshoot or under shoot for reasons of cargo safety, passenger comfort and to avoid accidental grounding in a narrow shipping lane like the one in the Hooghly in Calcutta port. A little longer response time is not a problem in these cases. So we see that, a position control system

should be designed differently to meet different specifications depending on its area of application.

We have observed in Section 6.4 of Chapter 6, how the variation of a single parameter like the forward path gain influences the transient response- obviously due to variation of the closed loop poles. The root locus method is a graphical tool to study the variation of closed loop poles of a system when only one parameter, normally the forward path gain, is allowed to vary. However, the locus of roots of the characteristic equation of a multi-loop system may also be investigated as in the case of a single-loop system. Since the root locus method provides graphical information, it may be used, specially in design stage of a controller, to obtain qualitative information regarding the stability and performance of the system.

We present below a step by step procedure for the root locus method for continuous system. We shall consider typical transfer functions of closed loop systems to illustrate the root locus techniques.

7.3 ROOT LOCUS TECHNIQUE

The root-locus method has been established as a useful tool for the analysis and design of linear time-invariant control systems. The root-locus diagram is essentially a graphical plot of the loci of the roots of the characteristic equation of a system as a function of a real parameter K which varies from $-\infty$ to $+\infty$. It gives an indication of the absolute stability and, to some extent, the relative stability of a control system with respect to the variation of the system parameter K. We shall confine our discussions to only positive values of K, since it can be readily extended to the case of negative values of K.

Consider a closed loop system

$$\frac{C(s)}{R(s)} = \frac{KG_1(s)}{1 + KG_1(s)\,H(s)} \tag{7.1}$$

where the forward path gain K is kept as a separate design parameter. The forward path gain may include the process gain together with the error amplifier gain. Let the loop transfer function G(s)H(s) for a second order system be given by,

$$G(s)H(s) = KG_1(s)H(s) = \frac{K(s+8)}{s(s+5)} \tag{7.2}$$

Therefore, the characteristic equation is obtained as

$$1 + KG_1(s)H(s) = 0 \quad \Rightarrow \quad s(s + 5) + K(s + 8) = 0 \tag{7.3}$$

The two open loop poles of the system are located at $p_1 = 0$ and $p_2 = -5$ and one finite zero is located at $z_1 = -8$ whereas the other zero is at ∞. The closed loop poles of the system are the roots of the characteristic equation (7.3) and depends on the value of K. When K = 0, the closed loop poles are identical with the open loop poles, but they are different for $K \neq 0$.

The roots of the above characteristic equation are calculated for various values of K and are shown in Table 7.1. Fig. 7.1 shows the loci of the roots as K is varied from K = 0 to K = ∞. We observe from the figure that the second order equation (7.3) gives rise to two branches of the loci-one being p_1BEDz1 and the other is p_2BCDF. It is to be noted that as the value of K is increased, the two branches, originating at the finite poles p_1 and p_2 approach each other and meets at the point B producing a double root there and breaks away to follow separate paths BE and BC. As K is further increased, the two branches approach each other and meets at the point D producing a double root there and moves on to follow separate paths Dz_1 and DF

towards zeros located at z_1 and infinity respectively. The points B and D are referred to as, respectively, *breakaway* and *breakin* points.

Table 7.1

Value of K	*Root located at s_1*	*Root located at s_2*
0	0	– 5
1	– 2	– 4
1.2020	– 3.0868	– 3.1152
2.000	– 3.5 + *j*1.9365	– 3.5 – *j*1.9365
5.0000	– 5 + *j*3.873	– 5 – *j*3.873
10.0000	– 7.5 + *j*4.8734	– 7.5 – *j*4.8734
20.0000	– 12.5 + *j*1.9365	– 12.5 – *j*1.9365
20.7980	– 12.885	– 12.913
25	– 10	– 20
∞	– 8	– ∞

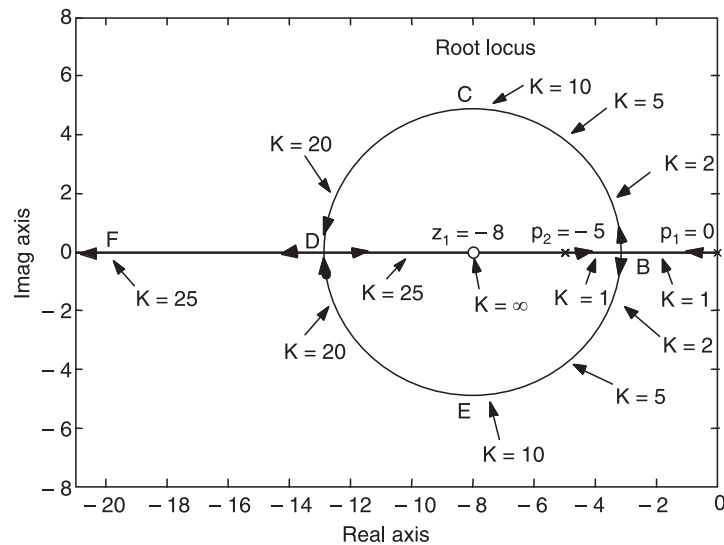



Fig. 7.1 Root loci of $KG_1(s)H(s) = \dfrac{K(s+8)}{s(s+5)}$

7.3.1 Properties of Root Loci

From the above observations and subsequent analysis we record below some important properties of the root loci of the characteristic equation for a control system with the loop transfer function $KG_1(s)H(s)$ and closed loop transfer function given by Equation (7.1).

The roots of the characteristic equation must satisfy the equation

$$1 + KG_1(s)\, H(s) = 0 \tag{7.4}$$

or $$G_1(s)H(s) = -\frac{1}{K} \tag{7.5}$$

In order to satisfy the Equation (7.5), the following two conditions must be satisfied simultaneously

(A) Magnitude Condition

$$|G_1(s)H(s)| = \frac{1}{|K|} \tag{7.6}$$

(B) Phase Angle Conditions

$K \geq 0$ $\quad \angle G_1(s)H(s) = (2k + 1)180°$

$= \text{odd multiples of } 180° \quad (7.7)$

$K \leq 0$ $\quad \angle G_1(s)H(s) = 2k\ 180°$

$= \text{even multiplies of } 180° \quad (7.8)$

where k is any integer *i.e.*, $k = 0, \pm 1, \pm 2, \ldots$.

It is to be noted that the phase angles are all measured in a counterclockwise direction from a horizontal line.

So, any point in the plane of the root that satisfy the conditions (7.6) and (7.7) or (7.8) simultaneously will be a point on the root loci of the characteristic equation (7.5). Of the above two conditions, the **conditions on phase angle are more critical** than that of the magnitude condition. We can always find a value of K (not necessarily an integer) that will satisfy the magnitude condition of (7.6). But the condition on phase, demands that any search point $s = s_s$ will be a point on the root loci if the phase angle is odd (K > 0) or even (K < 0) multiple of 180°. The construction of root loci, therefore, is essentially to search for a trial point $s = s_s$ that satisfy the phase conditions (7.7), for K > 0 or (7.8), for K < 0 and then find the value of K from the amplitude condition (7.6).

Based on these observations, the properties for root loci in the plane of the roots are recorded below for positive values of K. The relevant changes in the properties can be readily incorporated for negative values of K.

CAD tools like MATLAB may be used for drawing root loci. However, the engineer must be familiar with the rules in order to interpret the plots.

1. **Points of the loci corresponding to K = 0**

The points on the root loci corresponding to the value of K = 0 are at the poles of G(s)H(s). This may include the poles at infinity where the number of zeros is greater than that of the poles (occurring in a derived loop gain in some design approaches).

If p_j are the finite poles and z_i are finite zeroes of the loop gain $KG_1(s)H(s)$, the characteristic polynomial (7.4) can be written as

$$F(s) = 1 + KG_1(s)H(s) = 1 + K\frac{\prod_{i=1}^{m}(s + z_i)}{\prod_{j=1}^{n}(s + p_j)} = 0 \tag{7.9}$$

So, the magnitude condition (7.6) becomes

$$\frac{\prod_{i=1}^{m} |(s+z_i)|}{\prod_{j=1}^{n} |(s+p_j)|} = \frac{1}{K} \qquad (7.10)$$

Now as K approaches zero, the right hand side of Equation (7.10) becomes infinite which is satisfied when s approaches the poles $-p_j$.

2. **Points of the loci corresponding to K = ∞**

The points on the root loci corresponding to the value of K = ∞ are at the zeros of G(s)H(s), including those at the infinity. This is evident from Equation (7.10), as K approaches infinity, the right hand side approaches zero in value, which requires that s must approach the zeros $-z_i$. Again, if the denominator is of order higher than the numerator ($n > m$), then $s = \infty$, makes the left hand side zero corresponding to K = ∞. Therefore, some roots are located at $s = \infty$.

In Fig. 7.1, the root loci originates at the finite poles $p_1 = 0$ and $p_2 = -5$ for K = 0 and terminates at the finite zero at $s_1 = -8$ and at $s_2 = \infty$ for K = ∞.

3. **Number of branches of separate root loci**

The number of root loci is equal to the number of finite poles or zeros of G(s)H(s) whichever is greater. This is apparent, since the root loci must start at the poles and terminate at the zeros of G(s)H(s), the number of branches of loci is equal to the maximum of the two numbers finite poles or zeros.

In the system of Equation (7.2), since the number of finite poles is 2, which is higher than the finite zero, which is 1 in this case, the number of branches of root loci is 2 (see Fig. 7.1).

4. **Symmetry of root loci**

The root loci are symmetric with respect to the real axis, since the complex roots occur in complex conjugate pairs.

5. **Root loci on the real axis**

Root loci are found on a given section of the real axis of the s-plane only if the total number of real poles and real zeros of G(s)H(s) to the right of the section is odd for K > 0.

With reference to the distribution of poles and zeros of $G_1(s)H(s)$ as shown in Fig. 7.2, if we take the search point s_s, anywhere on the real axis between p_1 and p_2, the angular contribution of the complex conjugate poles p_3 and p_4 at the search point is 360°, (this is also true for any complex conjugate zeros, when present). The poles and zeros on the real axis to the right of s_s each contribute 180° with appropriate sign included. Therefore, with reference to Fig. 7.2, we can write the phase condition (7.7) as

$$\varphi_1 + \varphi_2 - [\theta_1 + \theta_2 + (\theta_3 + \theta_4) + \theta_5 + \theta_6] = (2k + 1)180° \qquad (7.11)$$

where φ_1, φ_2 are the phase angles contributed by the zeros z_1 and z_2 at the search point s_s, and θ_i, $i = 1$ to 6 are the phase angles contributed by the poles p_i, $i = 1$ to 6.

Computing the angles from Fig. 7.2, we have;

$$0° + 0° - [180° + 0° + (360°) + 0° + 0°] = (2k + 1)180°$$

or

$$-540° = (2k + 1)180°$$

This can be satisfied with $k = -2$.

This observation is true if the search point is taken anywhere on the stretch of the real axis lying between p_1 and p_2. Hence, the entire stretch of the real axis between p_1 and p_2 will be part of the root locus. In this way, we can show that the part of the real axis lying between p_5 and z_1, as well as between p_6 and z_2 are part of the root locus (vide Fig. 7.2).

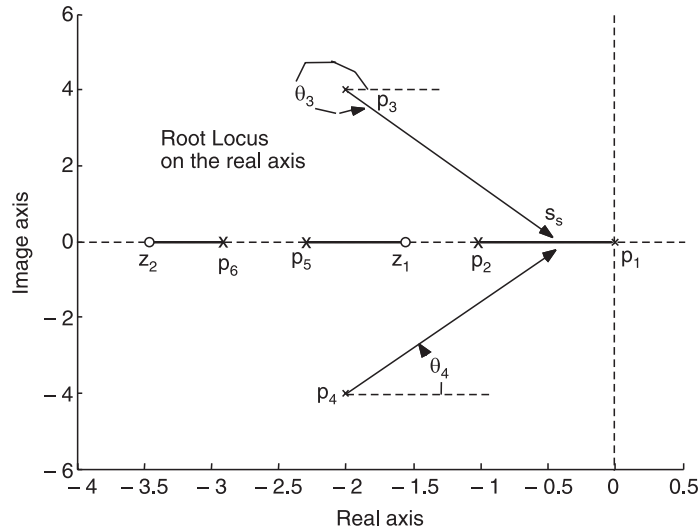



Fig. 7.2 Root loci on the real axis

6. **Asymptotes of root loci**

For large values of s the root loci are asymptotic to the straight lines with angles given by

$$\theta_k = \frac{(2k+1)180^\circ}{|n-m|} \tag{7.12}$$

where $k = 0, 1, 2, \ldots \mid n - m \mid - 1$; n is the number of finite poles of G(s)H(s), and m is the number of finite zeros of G(s)H(s).

Plotting the root loci are greatly facilitated if one can determine the asymptotes approached by the various branches as s takes on large values. Now from Equation (7.9) we can write

$$\lim_{s \to \infty} K \frac{\prod_{i=1}^{m} (s + z_i)}{\prod_{j=1}^{n} (s + p_j)} = -1$$

or

$$\frac{K}{s^{n-m}} = -1 \quad \Rightarrow \quad -K = s^{n-m} \quad \text{or} \quad |-K| = |s^{n-m}|$$

and

$$\angle -K = \angle s^{n-m} = (2k+1)180^\circ \quad \Rightarrow \quad \angle s = \theta_k = \frac{(2k+1)180^\circ}{n-m} \tag{7.13}$$

where θ_k is the contribution to phase angle by a finite pole or zero at a search point which is far away from them.

7. **Intersection of the asymptotes on the real axis**

The intersection of the asymptotes lies only on the real axis of the s-plane. The points of intersection of the asymptotes on the real axis is given by

$$\sigma_1 = \frac{\Sigma \text{ real parts of poles of } G(s)H(s) - \Sigma \text{ real parts of zeros of } G(s)H(s)}{n-m} \quad (7.14)$$

The center of the linear asymptotes, often called the **asymptote centroid,** is determined by considering the characteristic equation (7.9). For large values of s, only the higher-order terms need be considered, so that the characteristic equation reduces to

$$1 + \frac{Ks^m}{s^n} = 0 \quad \Rightarrow \quad 1 + \frac{K}{s^{n-m}} = 0$$

This approximate relation indicates that the centroid of $(n - m)$ asymptotes is at the origin, $s = 0$. However, a more general approximation is obtained if we consider a characteristic equation of the form

$$1 + \frac{K}{(s-\sigma_1)^{n-m}} = 0$$

with the centroid at σ_1. Expanding the denominator and retaining the first two terms of the above expression, we have

$$1 + \frac{K}{s^{n-m} - (n-m)\sigma_1 s^{n-m-1} + \ldots} = 0 \quad (7.15)$$

Again from Equation (7.9) we can write

$$1 + \frac{K \prod_{i=1}^{m} (s + z_i)}{\prod_{j=1}^{n} (s + p_j)} = 1 + \frac{K(s^m + b_1 s^{m-1} + \ldots + b_m)}{s^n + a_1 s^{n-1} + \ldots + a_n}$$

$$= 1 + \frac{K}{s^{n-m} + (a_1 - b_1)s^{n-m-1} + \ldots + R(s)/N(s)} \quad (7.16)$$

where $b_1 = \sum_{i=1}^{m} z_i$ and $a_1 = \sum_{j=1}^{n} p_j$, $R(s)$ is the residue and $N(s) = \prod_{i=1}^{m} (s + z_i)$

For large values of s the characteristic equation may be written by considering only the first two terms of relation (7.16)

$$1 + \frac{K}{s^{n-m} + (a_1 - b_1)s^{n-m-1}} \quad (7.17)$$

Now, equating the coefficients of s^{n-m-1} in relations (7.17) and (7.15) we obtain

$$(a_1 - b_1) = -(n-m)\,\sigma_1$$

Hence $\sigma_1 = -\dfrac{a_1 - b_1}{n-m}$, which is equivalent to Equation (7.14).

8. **Angles of departure from complex poles and angles of arrival at complex zeros**

The angle of departure of the root loci from a complex pole or the angle of arrival at a complex zero of G(s)H(s) can be determined by considering a search point s_s very close to the pole, or zero that satisfy the phase condition of relation (7.7)

For illustration, let us consider the loop gain $G(s)H(s) = \dfrac{K}{s(s+4)(s+2+j4)(s+2-j4)}$.

There are four finite poles located at $p_1 = 0$, $p_2 = -4$, $p_3 = -2 - j4$ and $p_4 = -2 + j4$, which are shown in Fig. 7.3. We take a search point $s = s_s$ very close to the complex pole $p_3 = -2 + j4$ and apply the conditions of phase angles as follows:

$$-\angle s_s - \angle(s_s + 4) - \angle(s_s + 2 + j4) - \angle(s_s + 2 - j4) = (2k + 1)180$$

or $$-\theta_1 - \theta_2 - \theta_3 - \theta_4 = (2k + 1)\,180°$$

Using the values from the graph, we have, $-135° - 45° - \theta_3 - 90° = (2k + 1)180°$

Therefore, $\theta_3 = -(2k + 1)180° - 270° = 540° - 270° = 270°$, for $k = -2$

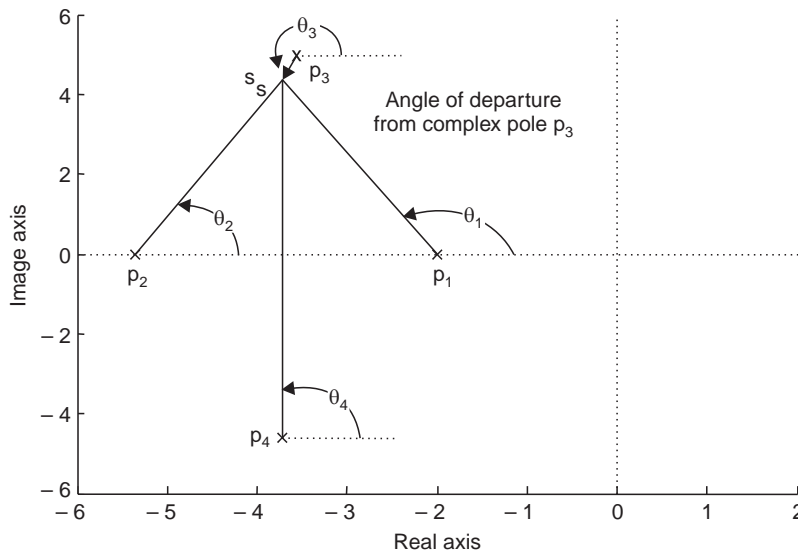



Fig. 7.3 Determination of angle of departure from complex poles

9. **Intersection of the root loci with the imaginary axis in the *s*-domain**

The value of K at the point of intersection of the root loci with the imaginary axis s = jω may be determined by using the Routh-Hurwitz test.

10. **Breakaway points**

The breakaway points on the root loci are points at which multiple-order roots lie (point B in Fig. 7.1).

We have noted that the locus starts at the poles with K = 0 and terminates at the finite zeroes or at ∞ with K = ∞.

We have also found in Fig. 7.1 that the part of the real axis between the pole $s = 0$ and the pole $s = -5$ is lying on the root loci. So the two branches of loci start at the poles with K = 0 and approach each other as K is increased until the two branches meet at point B in Fig. 7.1.

Any further increase of K will cause the roots **breakaway** from the real axis. Therefore, so far as the real axis loci is concerned, as we move from one pole on the real axis towards the other pole, the value of K gradually increases and reaches a maximum value and then decreases to zero at the other pole (see Fig. 7.1 and 7.4). Similarly, in case a portion of real axis lying between two finite zeros is a part of root locus, two branches of loci will **breakin** on the real axis and move towards the finite zeros as K approaches infinity. It is apparent, that starting from one finite zero lying on the real axis corresponding to infinite value of K and moving towards the other finite zero on the real axis where K is again infinite, the value of K will attain a minimum value in between the two real axis zeros (see Fig. 7.5). For computing the breakaway or breakin points, we can rearrange the characteristic equation to isolate the multiplying factor K such that it can be in the form $K = F_1(s)$, where $F_1(s)$ does not contain K. *The breakaway and breakin points are then determined from the roots of the equation obtained by setting* $\frac{dK}{ds} = 0$.

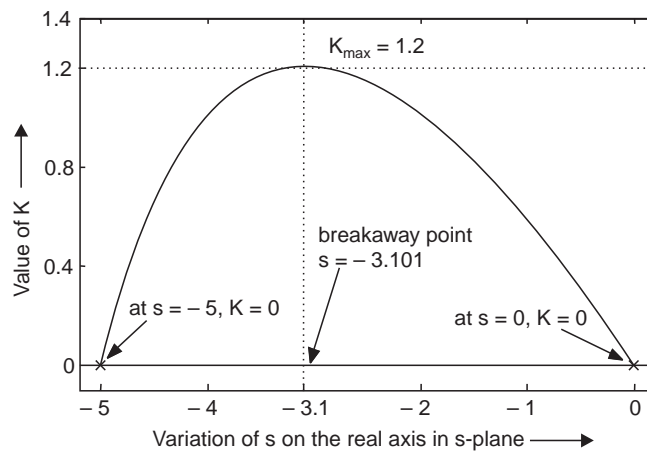



Fig. 7.4 The breakaway point of $K = -\frac{s(s+5)}{s+8} = F_1(s)$

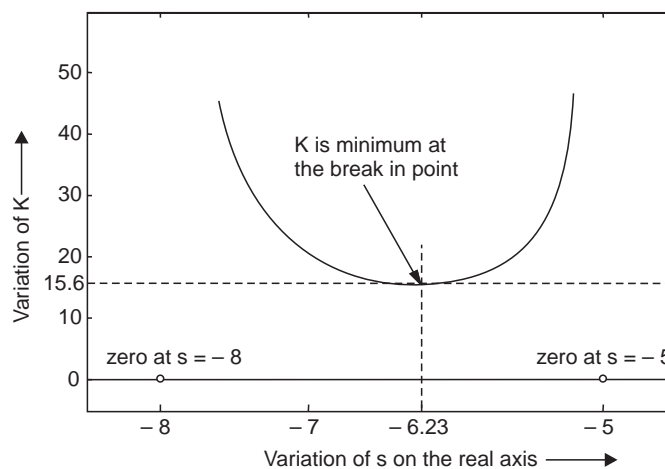



Fig. 7.5 The breakin point of $K = -\frac{(s^2 + 4s + 20)}{(s+5)(s+8)} = F_1(s)$

11. **Values of K on the root loci**

The value of K at any point s_1 on the root loci is determined from the following equation

$$|K| = \frac{1}{G(s_1)H(s_1)} \tag{7.18}$$

$$= \frac{\text{Product of lengths of vectors from } poles \text{ of } G(s)H(s) \text{ to } s_1}{\text{Product of lengths of vectors drawn from } zeros \text{ of } G(s)H(s) \text{ to } s_1}$$

With reference to Fig. 7.6, we draw the constant damping ratio line $\theta = \cos^{-1}(0.707)$ which meets the root locus of $KG_1(s) = \dfrac{K(s+5)}{(s+2+j4)(s+2-j4)}$ at $s_1 = -5 + j5$ and $s_2 = -5 - j5$. The value of K at the closed loop poles s_1 is found as (see Fig. 7.6) :

$$K = L_{p1} \,.\, L_{p2} / L_{z1} = (3.16)(9.5)/5 = 6.0$$

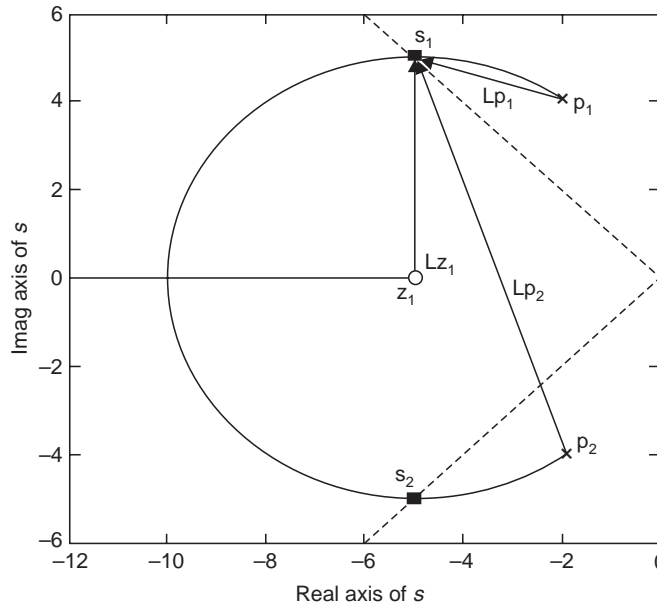



Fig. 7.6. Computation of K at a specified root-location s_1 on the locus diagram

7.4 STEP BY STEP PROCEDURE TO DRAW THE ROOT LOCUS DIAGRAM

In order to find the roots of the characteristic equation graphically on the plane of the root, we shall present a step by step procedure incorporating the properties listed above by considering some illustrative examples.

Example 7.1 Consider a control system with loop gain L(*s*) given by

$$L(s) = G(s)H(s) = \frac{K(s+6)}{s\,(s+2)\,(s+3)}$$

We are interested in determining the locus of roots for positive values of K in the range $0 \le K \le \infty$.

Step 1: Writing the characteristic equation using the loop gain.

The characteristic equation for this example is given by

$$1 + \frac{K(s+6)}{s\,(s+2)\,(s+3)} = 0 \tag{7.19}$$

In general the characteristic equation should be written using the loop gain L(*s*) as

$$F(s) = 1 + L(s) = 0$$

and the equation be rearranged, if required, such that the parameter of interest, K, appears as the multiplying factor in the form:

$$1 + KL_1(s) = 0 \tag{7.20}$$

Step 2: Write the polynomial in the factored form of poles and zeros as follows :

$$F(s) = 1 + L(s) = 1 + K\frac{\prod_{i=1}^{m}(s+z_i)}{\prod_{j=1}^{n}(s+p_j)} = 0 \tag{7.21}$$

For this example, poles and zeros are already factored. There are 3 finite poles located at $s_{p1} = 0$, $s_{p2} = -2$, $s_{p3} = -3$ and one finite zero located at $s_{z1} = -6$ and two zeros at infinity.

Step 3: Locate the poles and zeros on the *s*-plane on a graph paper.

We note that Equation (7.21) may be rewritten as

$$\prod_{j=1}^{n}(s+p_j) + K\prod_{i=1}^{m}(s+z_i) = 0. \tag{7.22}$$

For this example, we have $F(s) = s(s+2)(s+3) + K(s+6) = 0$

So, when K = 0, the roots of the characteristic equation are the poles of F(*s*), which was mentioned in the properties.

Also, when $K \to \infty$, the roots of the characteristic equation are the zeros of F(*s*). *The locus of the roots of the characteristic equation F(s) = 0, therefore, begins at the poles of F(s) and ends at the zeros of F(s) as K increases from 0 to infinity.*

So, the root locus starts from 0, –2, and –3 for K = 0 and terminates at –6, ∞, ∞ for K = ∞ (vide Fig. 7.7).

Step 4: Determine the number of separate loci.

In this example, there are three branches of the root loci, of which one terminates at the finite zero located at $s_{z1} = -6$ while two branches terminate at + ∞. In general, with *n* finite poles and *m* finite zeros and $n > m$, we have $N = n - m$ *branches of the root locus approaching the N zeros at infinity.*

Step5: The *root loci is symmetrical with respect to the horizontal real axis* because the complex roots must occur as complex conjugate pairs.

Step 6: Locate the segments of the real axis that are part of root loci.

The section of the real axis will be part of the root locus if the count of real axis poles and zeros looking to the right is an odd number.

So, the portion of the real axis lying between – 6 and – 3 and between 0 and – 2 are part of root loci.

Step 7: Asymptotes:

In this example, the number of finite pole, $n = 3$, and number of finite zero, $m = 1$, so that $n - m = 2$ branches of loci approach the two zeros at infinity asymptotically and makes angles θ_k. Therefore, from Equation (7.12) we find $\theta_0 = 90°$, for $k = 0$ and $\theta_1 = 270°$, for $k = 1$. In general, if the number of finite zeros of F(s) are m and the number of finite poles are n, then $N = n - m$ branches of root loci terminate at zeros at infinity as K approaches infinity. These branches approach the zeros at infinity **asymptotically** and they meet the real axis at σ_1, which is computed, for this example, as:

$$\sigma_1 = \frac{\Sigma \text{ real parts of poles of } G(s)H(s) - \Sigma \text{ real parts of zeros of } G(s)H(s)}{n - m} \quad (7.23)$$

$$= \frac{(0 - 2 - 3) - (-6)}{3 - 1} = 0.5$$

Step 8: Determine the point at which the locus crosses the stability boundary - the imaginary axis for continuous system.

The actual point at which the root locus crosses the imaginary axis is found with the help of Routh table (vide Section 5.2 in Chapter 5).

For this example, the characteristic equation is given by

$$F(s) = s(s + 2)(s + 3) + K(s + 6) = s^3 + 5s^2 + (6 + K)s + 6K = 0$$

The Routh array becomes

s^3	1	6 + K
s^2	5	6K
s^1	$\frac{5K + 30 - 6K}{5}$	0
s^0	6K	0

From the Routh table, the conditions of stability are:

(1) $6K > 0$, implying $K > 0$

(2) $30 - K > 0$, implying $K < 30$

At K = 30, the roots lie on the imaginary axis and their values are obtained from the auxiliary equation $5s^2 + 6K = 0$, yielding a solution of $s = \pm j\omega = \pm j6$.

Step 9: Determine the breakaway point on the real axis (if any).

For the present case, we can write

$$K = -\frac{s(s + 2)(s + 3)}{s + 6} = F_1(s) \quad (7.24)$$

Now

$$\frac{dK}{ds} = \frac{-2s^3 - 23s^2 - 60s - 36}{(s + 6)^2}$$

Setting the above derivative to zero, we get $s_1 = -8.0517$, $s_2 = -2.5827$ and $s_3 = -0.8656$ of which $s_3 = -0.8656$ is the solution with a corresponding K = 0.408, whereas the other two solutions corresponds to negative values of K, respectively, as can be verified by substitution in Equation (7.24).

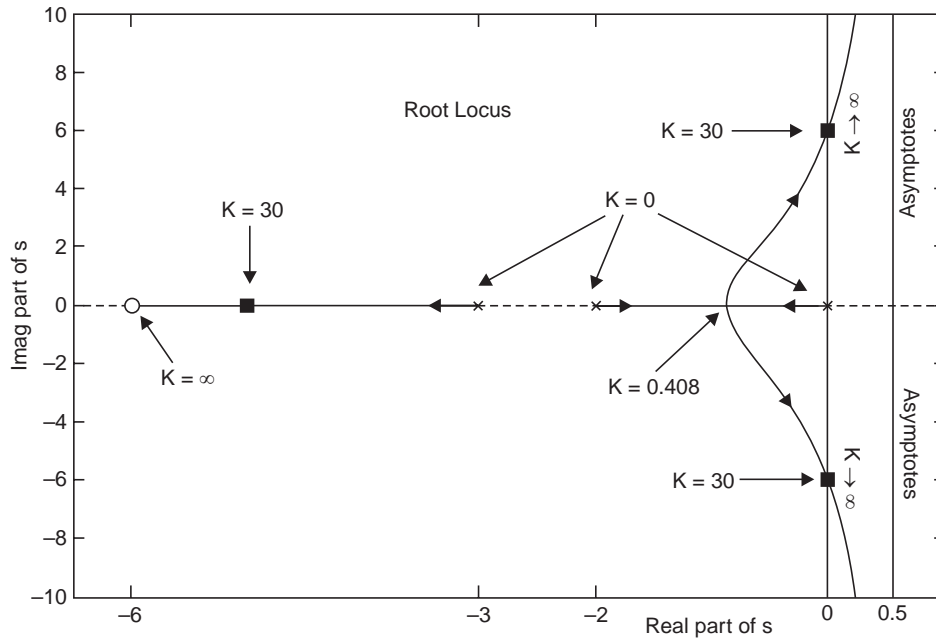



Fig. 7.7 Root loci of $G(s)H(s) = \dfrac{K(s+6)}{s\,(s+2)\,(s+3)}$

Step 10: Determine the angles of departure from complex poles and angles of arrival at complex zeros of the root locus. This rule is not applicable for this example, since all the open loop poles and zeros are real.

Step 11: We can determine the parameter value K_s at a specific root s_s on the locus using the magnitude requirement of relation (7.6). The magnitude requirement at s_s is

$$K_s = \frac{\prod_{j=1}^{n} |(s_s + p_j)|}{\prod_{i=1}^{m} |(s_s + z_i)|} \tag{7.25}$$

The parameter K_s can also be computed graphically as follows:

$$K_s = \frac{\prod_{j=1}^{n} L_j}{\prod_{i=1}^{m} L_i} \tag{7.26}$$

where L_j and L_i are vector lengths of the search point s_s measured from pole p_j and zero z_i. The complete locus, based on the above data, may be drawn as shown in Fig. 7.7. It is to be noted that all the steps discussed above may not be executed for a given problem. For instance, step 10 was not needed for drawing the root loci in Fig. 7.7 but will be necessary in the next example.

Example 7.2 We consider the loop gain $G(s)H(s) = \dfrac{K}{s(s+2)(s+2+j4)(s+2-j4)}$

and follow the step by step procedure to obtain the root locus diagram for $0 < K < \infty$.

Step 1: Characteristic equation: The characteristic equation is given by

$$1 + \frac{K}{s(s+2)(s+2+j4)(s+2-j4)} = 0$$

Step 2: Poles and zeros of loop gain : There are four finite poles for the loop gain located at $p_1 = 0,\ p_2 = -2,\ p_3 = -2 + j4$ and $s_4 = -2 - j4$ and no finite zero such that $n = 4$, $m = 0$ and $N = n - m = 4$.

Step 3: The poles are located on a graph paper as shown in Fig. 7.8.

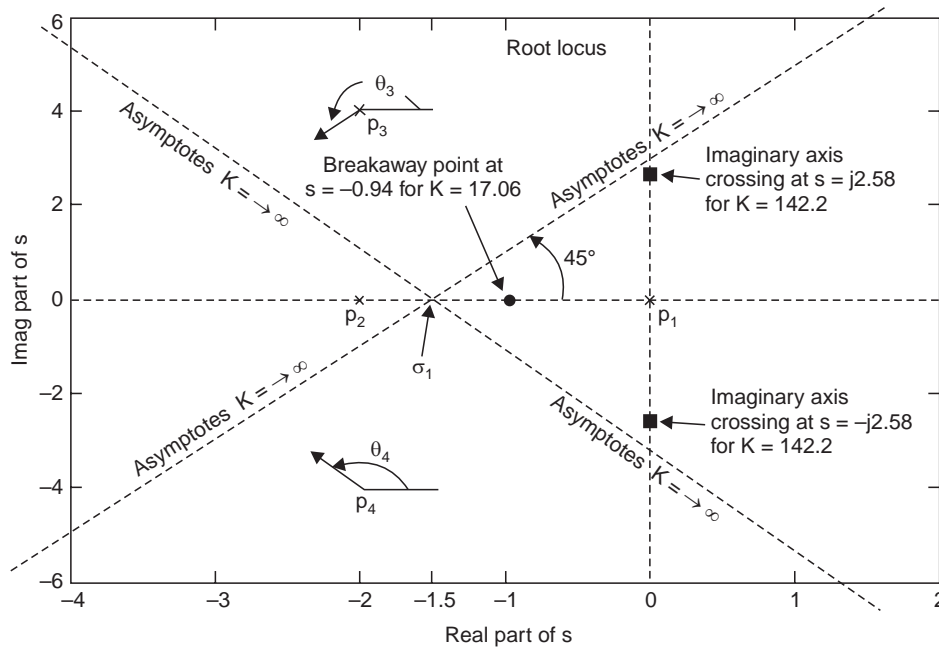



Fig. 7.8. Illustration of the steps for root locus plot for Example 7.2

Step 4: Number of branches of the root loci: Since, maximum $(n, m) = 4$, there will be 4 branches of root loci and all four will terminate at 4 zeros located at infinity.

Step 5: The root loci is symmetrical with respect to the horizontal real axis because the complex roots must occur as complex conjugate pairs.

Step 6: Real axis root locus:

For this example, the portion of the real axis lying between – 2 and 0 is part of root loci.

Step 7: Asymptotes: The angles of the asymptotes are found from the relation

$$\theta_k = \frac{(2k+1)180°}{n-m} = \frac{(2k+1)180°}{4}$$

Therefore, we have $\theta_0 = 45°$, $\theta_1 = 135°$, $\theta_2 = 225°$ and $\theta_3 = 315°$ corresponding to $k = 0, 1, 2,$ and 3 respectively.

The asymptotes intersect on the real axis at σ_1(see Fig. 7.8) and are found as

$$\sigma_1 = \frac{0-2-2-2}{4} = -1.5$$

Step 8: Points at which the locus crosses the imaginary axis.

For this example, the characteristic equation is given by

$$F(s) = s^4 + 6s^3 + 28s^2 + 40s + K = 0$$

The Routh table for the above characteristic equation is shown below:

s^4	1	28	K
s^3	6	40	0
s^2	64/3	K	0
s^1	a	0	0
s^0	K	0	0

where $a = 40 - (9/32)K$

For stability the following conditions must be satisfied:

(1) $K > 0$

(2) $K < 142.2222$

The roots will lie on the imaginary axis for K = 142.222 and the imaginary axis roots are found from the auxiliary equation $(64/3)s^2 + K = 0$. The values of s are found to be $s = \mp j\omega = \mp j2.58$.

Step 9: Breakaway points: From step 8 the characteristic equation is rearranged as:

$$K = -(s^4 + 6s^3 + 28s^2 + 40s) = F_1(s)$$

Therefore, we find $\dfrac{dK}{ds} = (-4s^3 - 18s^2 - 56s + 40)$, and by setting this derivative to zero, we get the following values of breakaway points:

$$s = -0.9382,$$

$$s = -1.7809 + j2.7362$$

and $$s = -1.7809 - j2.7362$$

The value of $s = -0.9382$ is the only permissible solution for the breakaway point lying on the real axis and the corresponding value of K is found to be K = 17.0620 by substituting $s = -0.9382$ in the characteristic equation. The other solutions for the breakaway points are not acceptable as they produce complex values of K.

Step 10: The angles of departure from complex poles.

By taking the search point s_s very close to the complex pole $p_3 = -2 + j4$ and following the procedure illustrated in Fig. 7.3, will produce an angle of departure θ_3 of 225°. Similarly, by taking a search point very close to the complex pole $p_4 = -2 - j4$, the angle of departure θ_4 is found to be 135° (see Fig. 7.8).

Using the above information we can draw the complete root loci diagram shown in Fig.7.9.

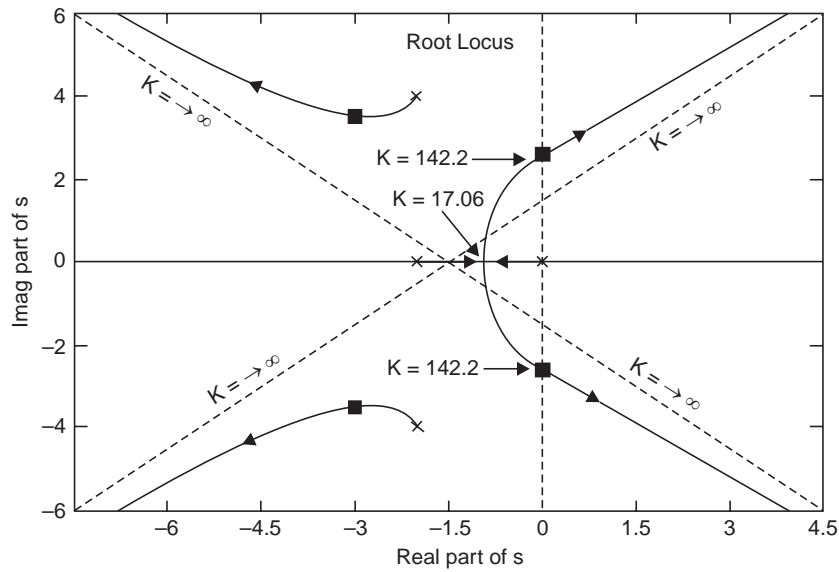



Fig. 7.9 Root locus diagram for $G(s)H(s) = \dfrac{K}{s(s+2)(s+2+j4)(s+2-j4)}$

Example 7.3 We now consider the following transfer function of a system to illustrate the root loci plot

$$F(s) = s(s + 2)(s^2 + 8s + 80) + K(s + 5) = 0$$

We divide the above equation by the factors not involving K to get

$$1 + \frac{K(s+5)}{s(s+2)(s^2+8s+80)} = 0$$

Step 1: The equation above is the characteristic equation.

Step 2: The finite poles are at $p_1 = 0$, $p_2 = -2$, $p_3 = -4 + j8$ and $p_4 = -4 - j8$ and one finite zero is located at $z_1 = -5$. So we have $n = 4$, $m = 1$ and $N = n - m = 3$.

Step 3: The poles and the zero are located on a graph paper as shown in Fig. 7.10.

Step 4: Since, maximum $(n, m) = 4$, there will be 4 branches of root loci and three will terminate at 3 zeros located at infinity and one branch will terminate at $z_1 = -5$.

Step 5: The root loci is symmetrical with respect to the horizontal real axis.

Step 6: The portion of the real axis lying between 0 and – 2 is part of root loci.

Step 7: The angles of the asymptotes are found from the relation:

$$\theta_k = \frac{(2k+1)\,180}{n-m} = \frac{(2k+1)\,180}{4}$$

Therefore, we have $\theta_0 = 45°$, $\theta_1 = 135°$, $\theta_2 = 225°$ and $\theta_3 = 315°$, corresponding to $k = 0, 1, 2$, and 3 respectively. The asymptotes intersect on the real axis at σ_1 given by

$$\sigma_1 = \frac{0 - 2 - 4 - 4 - (-5)}{4 - 1} = -1.67$$

Step 8: For this example, the characteristic equation is simplified as:

$$F(z) = s^4 + 10s^3 + 96s^2 + (160 + K)s + 5K = 0$$

Therefore, the Routh-Hurwitz array is obtained as :

s^4	1	96	5K
s^3	10	160 + K	0
s^2	$\dfrac{800 - K}{10}$	5K	0
s^1	A	0	0
s^0	5K	0	0

where $$A = \frac{128000 + 140\,K - K^2}{800 - K}$$

So, we have the following conditions for stability:

(1) $K < 800$

(2) $K^2 - 140K - 128000 < 0$

(3) $K > 0$

For finding the values of K for which the loci crosses the imaginary axis in the s-plane, the inequality in condition (2) be replaced by equality to get the positive values of K = 434.55 for which the root loci will cross the imaginary axis. Putting this value of K in the auxiliary equation corresponding to the s^2 row in the Routh-Hurwitz table we get:

$$s^2 + 59.4541 = 0$$

This gives the frequencies at which the root loci crosses the imaginary axis as

$$s = \pm j\omega = \pm j7.7106$$

Step 9: The breakaway(breakin) points are given as the solution of

$$\frac{dK}{ds} = \frac{-(3s^4 + 40s^3 + 246s^2 + 960s + 800)}{(s+5)^2} = 0$$

which are $s_1 = -1.0869$, $s_2 = -7.4466$, $s_3 = -2.399 + j5.2141$ and $s_4 = -2.399 - j5.2141$

The solutions s_1 and s_2 satisfy the phase conditions for positive values of K and the corresponding values of K are 18.4 and 1256 respectively, whereas s_3 and s_4 do not satisfy the phase conditions for real values of K and are rejected. We note that the point on the root locus corresponding to K = 1256 is a *break-in* point between a zero at $-\infty$ and a finite zero at -5.

Step 10: The angles of departure from complex poles p_3 and p_4 are found as was illustrated in Fig. 7.3.

These are given by: $\theta_3 = 274.4°$ and $\theta_4 = 85.6°$

Using the above information, the complete root loci may be drawn for K > 0 and is shown in Fig. 7.10.

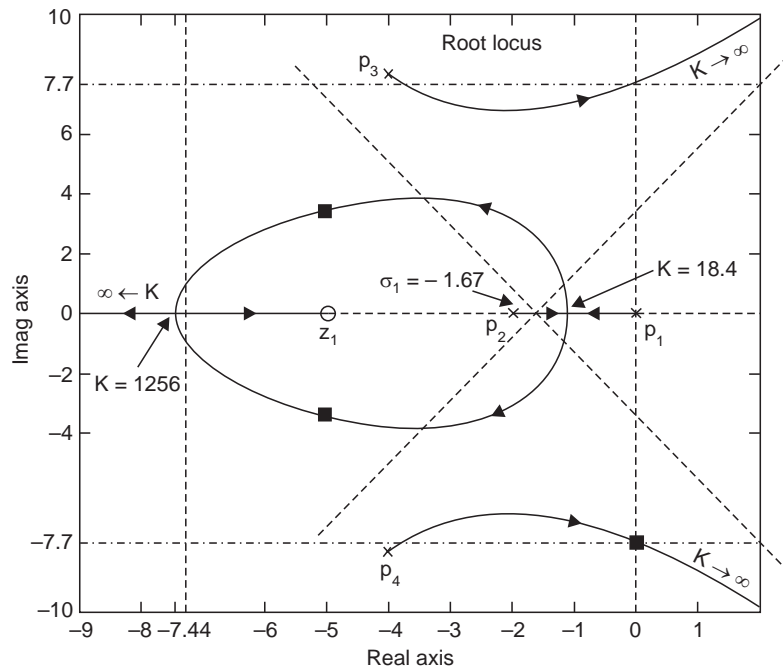



Fig. 7.10 Root loci of $F(s) = s(s + 2)(s^2 + 8s + 80) + K(s + 5) = 0$

Example 7.4 We now consider another loop transfer function for illustrating the root locus plot.

Let
$$G(s)H(s) = \frac{K}{s(s+2)(s+1-j3)(s+1+j3)}$$

Step 1: The characteristic equation is given by:

$$1 + \frac{K}{s\,(s+2)(s+1-j3)(s+1+j3)} = 0$$

Step 2: There are four finite poles for the loop gain located at $s_{p1} = 0$, $s_{p2} = -2$, $s_{p3} = -1 + j3$ and $s_{p4} = -1 - j3$ and no finite zero. So we have $n = 4$, $m = 0$ and $N = n - m = 4$.

Step 3: The poles are located on a graph paper as shown in Fig. 7.11

Step 4: Since, maximum $(n, m) = 4$, there will be 4 branches of root loci and all four will terminate at 4 zeros located at infinity.

Step 5: The root loci is symmetrical with respect to the horizontal real axis.

Step 6: The portion of the real axis lying between – 2 and 0 is part of root loci.

Step 7: The angles of the asymptotes are found from the relation (7.12).

Therefore, we have $\theta_0 = 45°$, $\theta_1 = 135°$, $\theta_2 = 225°$, and $\theta_3 = 315°$, corresponding to k = 0, 1, 2 and 3 respectively.

The asymptotes intersect on the real axis at σ_1 given by

$$\sigma_1 = \frac{0-2-1-1}{4} = -1$$

Chapter 7

Step 8: Points at which the locus crosses the imaginary axis in the s-plane.

For this example, the characteristic equation is given by

$$F(z) = s^4 + 4s^3 + 14s^2 + 20s + K = 0$$

Therefore, the Routh-Hurwitz array is

s^4	1	14	K
s^3	4	20	0
s^2	9	K	0
s^1	A	0	0
s^0	K	0	0

where $A = \dfrac{180 - 4K}{9}$

From the first column of the Routh-Hurwitz Table, we find the following conditions for stability:

(1) $K > 0$

(2) $K < 45$

For finding the values of K for which the loci crosses the imaginary axis in the s-plane, the inequality in condition (2) be replaced by equality to get value of K = 45 for which the root loci will cross the imaginary axis. Putting this value of K in the auxiliary equation corresponding to the s^2 row in the Routh-Hurwitz table we get:

$$9s^2 + 45 = 0$$

This gives the frequencies at which the root loci crosses the imaginary axis as

$$s = \pm j\omega = \pm j2.2360$$

Step 9: The breakaway points are given as the solution of

$$\frac{dK}{ds} = -(4s^3 + 12s^2 + 28s + 20) = 0$$

which are $s_1 = -1$, $s_2 = -1 + j2$ and $s_3 = -1 - j2$

All the above three values of s are breakaway points since they satisfy the phase conditions. The corresponding values of K are 9 and 25 respectively.

Step 10: The angles of departure from complex poles s_3 and s_4 are found as before.

These are given by: $\theta_3 = 90°$ and $\theta_4 = 90°$

Using the above information we draw the complete root loci shown in Fig. 7.11.

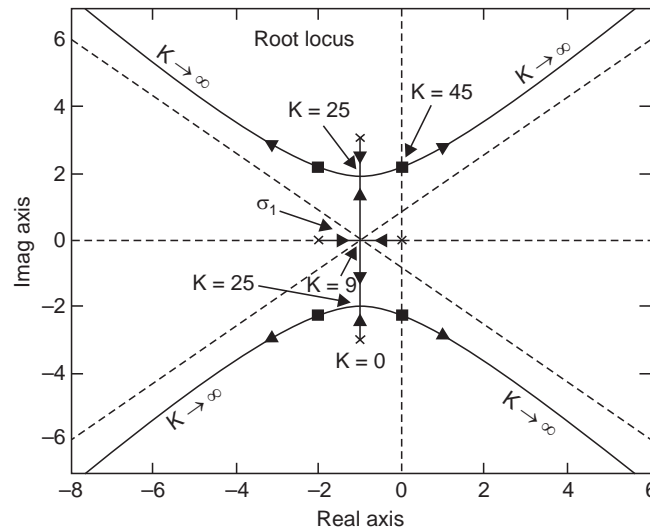



Fig. 7.11 Root locus of $G(s)H(s) = \dfrac{K}{s(s+2)(s+1-j3)(s+1+j3)}$

7.5 ROOT LOCUS DESIGN USING GRAPHICAL INTERFACE IN MATLAB

The root locus is a powerful design tool of control systems in classical approach. Since, by varying the forward path gain K, the location of the closed loop poles can be varied, it affects the overshoot, rise time, settling time as well as steady state error. However, we can not set all the above performance measures to specified values simultaneously by varying only a single parameter K. The ability to change K along with the freedom of introducing compensator poles and zeros (vide Chapter 8) in the structure of control systems, greatly enhances the capability of a control engineer to meet more than one design specifications simultaneously. The design of compensators using root locus technique has been taken up in Section 8.3.1 of Chapter 8. In the example below, we shall discuss the use of *rltool* in MATLAB for the selection of K to meet a prescribed constraint on overshoot, settling time, and natural frequency of a closed loop system from its root locus diagram.

Example 7.5 Draw the root locus diagram of a plant with loop gain $G(s)H(s)$ given below, and set the value of K so that the step response exhibits an overshoot less than 15%, when the loop is closed.

$$G(s)H(s) = K(s + 6) / s(s + 4)(s + 8);$$

In the MATLAB command window enter >> gh = zpk([–6], [0 –4 –8],1);

Enter >> *rltool,*

The GUI **SISO Design for System Feedback config** opens with the following feedback configuration:

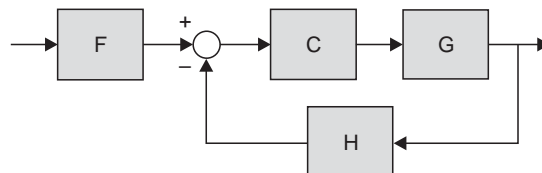



Fig. 7.12 Feedback configuration in MATLAB GUI rltool

Go to **File** and from the drop down menu click **Import** and select *gh* and transfer to **G** by clicking the arrow near **G.** Leave F = 1, C = 1 and H = 1 unchanged. Click **OK** to see the root locus diagram. By clicking right mouse on the root locus window, bring pointer on **Design Constraints** and select **New** which opens a window **New Constraints.** Now from the drop down menu of **Constraint type,** select **Percent Overshoot**. Select value 15 and click OK. You will see two lines of constant damping ratio δ as shown in Fig. 7.13. Now bring the pointer over the rectangular marker on the locus when the pointer becomes a hand symbol. Drag the hand pointer with left mouse pressed, along the locus of increasing K and stop short of constant damping ratio line. Note the value of K from the top left corner inside the box **Current Compensator.** If required, type in the box finer values of K. For the example at hand, we get a value of K = 25. You can check the step response with the value thus set by pressing **Analysis** at the top of the window and clicking **Response to Step Command** and note that the percent overshoot (14.2% in the present case) by pressing right mouse and choosing **Characteristics** and pressing **Peak Response.** Alternatively, one can find the step response of the closed loop control system from the command window. In a similar manner, we can choose the value of K for other types of constraints, like **Settling Time** and **Natural Frequency** from the constraint window. It is apparent that arbitrary values of settling time and natural frequency can not be realized by setting K alone for given system. For instance, a settling time of t_s less than 1.56 sec (at K = 198) and natural frequency of greater than 8.0 (at K = 0) can not be attained for the system in Example 7.3 by changing K alone.

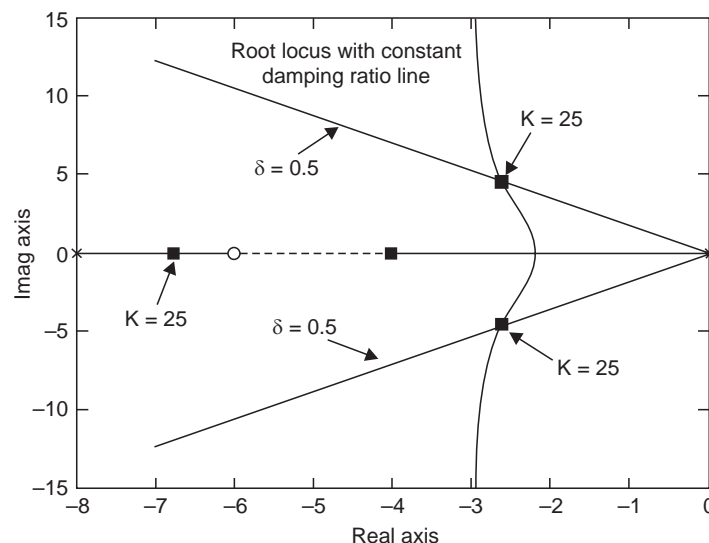



Fig. 7.13 Selection of gain K for a prescribed overshoot

7.6 ROOT LOCUS TECHNIQUE FOR DISCRETE SYSTEMS

All the properties mentioned in Section 7.3.1 in connection with the root locus plot of the continuous systems, are equally applicable to the root locus plot of discrete system with the exception of the rule number 9 related to the crossing of boundary of stable zone. In Section 10.14 in Chapter 10, we have shown that the boundary of stability for discrete system is the unity circle in the plane of the root. So, the relevant property should now be recast as shown below:

9. Intersection of the root loci with the unit circle (imaginary axis in the case of s-domain root loci)

The value of K at the point of intersection of the root loci with the unit circle $|z| = 1$ may be determined by using any stability tests like that of Jury-Blanchard test discussed in Sec 10.14.1 (*Routh-Hurwitz test in case of continuous* system).

So, considering a discrete system closed loop system

$$\frac{C(z)}{R(z)} = \frac{K G_1(z)}{1 + K G_1(z) H(z)} \tag{7.27}$$

all the step by step procedure in Section 7.4 can be repeated for the digital case, with the above amendment. As in the analog case, the forward path gain K is kept as a separate design parameter and it may include the process gain together with the error amplifier gain and $G(z)H(z)$ is the loop transfer function.

As expected, the MATLAB GUI *rltool* is equally applicable for root locus plot for discrete system. However, locus of constant δ and constant natural frequency undergoes a change because of the nonlinear relation between *s* and *z* (vide Section B.1.4 in Appendix B).

The continuous transfer functions can be converted to discrete domain by the MATLAB command *gdig* = *c*2*d* (*gh*, T), where *gh* is the continuous transfer functions, T is the sampling period. The root locus plot of any of the continuous transfer functions considered in the examples in Section 7.4 can be obtained by entering in MATLAB command window >> rlocus (*gdig*) and the gain K can be selected from the *rltool* for prescribed constraints. However, it is to be noted that the damping ratio in the *z*-domain is the equivalent damping ratio of the dominant roots in the *s*-plane.

7.7 SENSITIVITY OF THE ROOT LOCUS

The breakaway points were computed as the solutions of $\frac{dK}{ds} = 0$ for the continuous control system. Since the roots of the characteristic equation determines the time domain response, the pole sensitivity as a function of forward path gain is very important in closed loop system design. The root sensitivity for continuous control system may be defined as :

$$S_k = \frac{ds/s}{dK/K} = \frac{K}{s}\frac{ds}{dK} \tag{7.28}$$

At the breakaway points $\frac{dK}{ds} = 0$, so the root sensitivity S_k is infinite at the breakaway points. Therefore, care must be exercised in choosing the value of the forward path gain K for a predictable behavior of the closed loop system such that the root sensitivity is very small at the desired location.

MATLAB SCRIPTS

```
%Script_Fig7.7
% Root locus diagram of Example 7.1
clear all; close all hidden;
gh = zpk( [-6], [0 –2 –3],1);
rlocus(gh);
```

```
% Alternatively, after entering gh, type >> rltool in the command
% window to invoke the root locus tool. In the open rltool GUI, select
% the Import item from the File menu and import gh from work space
% in the forward path and click OK to get the root locus diagram
% with a marker. By dragging the marker with pressed left mouse
% one can vary the gain K on the locus diagram.
```

```
%Script_Fig7.9
% Root locus diagram of Example 7.2
clear all; close all hidden;
gh = zpk ( [], [0 –2 –2 + 4i –2 –4i],1);
rlocus(gh);
```

```
%Script_Fig7.10
% Root locus diagram of Example 7.3
clear all; close all hidden;
gh=zpk([-5],[0 –2 –4+8i –4-8i],1);
rlocus(gh);
```

```
%Script_Fig7.11
% Root locus diagram of Example 7.4
clear all; close all hidden;
gh=zpk([],[0 –2 –1+3i –1–3i],1);
rlocus(gh);
```

REVIEW EXERCISE

RE7.1 Draw the root locus for the following loop transfer functions for positive values of K:

(*a*) $G(s)H(s) = \dfrac{K}{s\,(s+2)^2}$ (*b*) $G(s)H(s) = \dfrac{K}{(s^2+5s+6)\,(s+4)}$

(*c*) $G(s)H(s) = \dfrac{K\,(s+3)}{s\,(s+1)\,(s+2)}$ (*d*) $G(s)H(s) = \dfrac{K\,(s^2+6s+13)}{s^2\,(s+2)}$

RE7.2 A unity feedback control system has a forward path transfer function given by:

$$G(s) = \frac{K\,(s^2+4s+8)}{s^2\,(s+1)}$$

Find the gain K such that the damping ratio of the complex roots is given by $\delta = 0.65$. Also verify that for this K the dominant roots are given by $s = -2.12 \pm j2.46$.

Ans: K = 13.4

RE7.3 Consider a feedback system with a loop transfer function

$$G(s)H(s) = \frac{K}{(s+1)(s+6)(s+10)}$$

(*a*) Find the breakaway point on the real axis and the corresponding value of K.

(*b*) Also find the point of intersection of the asymptotes on the real axis.

Ans: (*a*) $s_1 = -3.06$, K = 42.05 (*b*) $\sigma_c = -5.67$

RE7.4 A mirror segment in a Telescope is actively controlled by a unity feedback system with G(*s*) given by :

$$G(s) = \frac{K}{s(s^2 + 2s + 6)}$$

(*a*) Obtain the root locus diagram (*b*) Find the gain when two roots lie on the imaginary axis.

Ans: (*b*) K = 12, $s_{1,2} = \pm j2.45$

RE7.5 A robot force control system with unity feedback has forward path transfer function :

$$K(s)G(s) = \frac{K(s+3)}{(s^2 + 2s + 4)(s^2 + 6s + 25)}$$

(*a*) Draw the root locus diagram and find the gain K that results in dominant roots with a damping ratio of 0.7. (*b*) Find the actual percent overshoot and peak time for the gain K found in part (*a*).

Ans: (*a*) K = 28.6 (*b*) over shoot 48%, $t_p = 1.13$ sec

RE7.6 A unity feedback system with forward path transfer function G(*s*) has a zero in the numerator as a design parameter as shown below:

$$G(s) = \frac{5(s+a)}{s(s+2)(s+4)}$$

(*a*) Draw the root locus as *a* is varies from 0 to 20.

(*b*) Using the root locus, estimate the percent over-shoot and settling time (2% criterion) of the system at *a* = 1, 2.5 and 5 for a step input.

(*c*) Find the actual overshoot and settling time at *a* = 0.5, 2.5 and 5.

Ans: (*c*) No over shoot, $t_s = 16.3$ sec; over shoot = 4.89%, $t_s = 2.92$ sec; over shoot = 29.9%, $t_0 = 4.59$ sec;

Hints: First we shall write the characteristic equation in the form 1 + *a* F(*s*) = 0 such that we can use root locus diagram of F(*s*) to find the effect of the variation of the parameter *a*. The characteristic equation is found to be $s(s+2)(s+4) + 5s + 5a = 0 \rightarrow 1 + \frac{5a}{s(s+2)(s+4)+5s} = 0.$

Therefore, drawing the root locus diagram of $\frac{5}{s(s+2)(s+4)+5s} = \frac{5}{s(s^2+6s+13)}$ will enable us to study the effect of variation of a

RE7.7 A negative unity feedback system has a plant transfer function

$$G(s) = \frac{Ke^{-sT}}{s+2}$$

where T = 0.2 second. The time delay may be approximated as: $e^{-sT} \cong \frac{\left(\frac{2}{T} - s\right)}{\left(\frac{2}{T} + s\right)}$

Chapter 7

So, replacing $e^{-0.2s}$ by $\frac{10-s}{10+s}$, obtain the root locus for the system for K > 0. Also find the range of K for which the system will be stable.

Ans: K < 12; Hints: G = *zpk* ([ ], [ ],1,'inputdelay',0.2)

RE7.8 A control system as shown in Fig. RE7.8 has a plant

$$G(s) = \frac{1}{s(s-2)}$$

(*a*) Draw the root locus diagram and show that the system is always unstable if $G_c(s) = K$. (*b*) Now draw the root locus diagram to find the range of $K \geq 0$ for which the system is stable when $G_c(s)$ is given by :

$$G_c(s) = \frac{K(s+2)}{(s+10)}$$

Also find the value of K when the locus crosses the imaginary axis along with the imaginary axis roots.

Ans: (*b*) K > 26.67; K = 26.67, $s_{1,2} = \pm j2.58$

Fig. RE 7.8 Feedback system

RE7.9 A unity feedback system has a plant

$$G(s) = \frac{K(s+0.8)}{s(s-2)(s+6)}$$

(*a*) Determine the range of K for stability. (*b*) Sketch the root locus. (*c*) Determine the maximum damping ratio δ of the stable complex roots.

Ans: (*a*) K > 15 ; (*c*) δ = 0.308.

RE7.10 The block diagram in Fig. 7.10 represents a system for automatically controlling spacing between vehicles on a guide way . The vehicle dynamics is represented by transfer function G(*s*) given by :

$$G(s) = \frac{(s+0.2)(s^2+2s+101)}{s(s-0.3)(s+0.8)(s^2+1.2s+256)}$$

(*a*) Draw the root locus of the system (*b*) Find all the roots when the loop gain $K = K_1K_2K_3 = 2000$.

Ans. (*b*) $r_1 = -27.7$, $r_2 = -9.84$, $r_3 = -2.3$, $r_4 = -0.607 \pm j17$, $r_5 = -0.341 \pm j0.133$

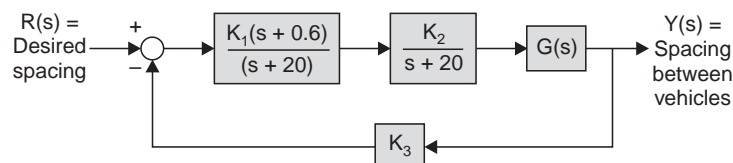



Fig. RE7.10 Guided vehicle control

PROBLEMS

P7.1 The control system for the heading of an aircraft is shown in Fig. P7.1. Find the maximum gain of the system for stable operation.

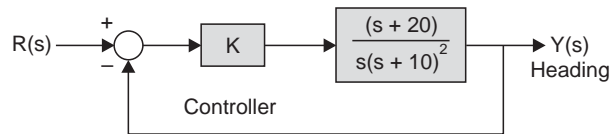



Fig. P7.1 Aircraft heading control

P7.2 Find the characteristic equation of the system with the matrix given by :

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -1 & -k & -2 \end{bmatrix}$$

Draw the root locus and find the range of k for which the system is stable.

P7.3 The block diagram of an Arc welding system with a vision system in the feedback path to monitor the diameter of molten metal puddle is shown in Fig. P7.3. This system uses a constant rate of feeding the wire to be melted for maintaining desired weld-quality.

(*a*) Calculate the maximum value for K for which the system will be in a stable state.

(*b*) With K set to 50% of its value found in (*a*) determine the roots of the characteristic equation.

(*c*) Estimate the overshoot of the system in (*b*) when it is subjected to a step input.

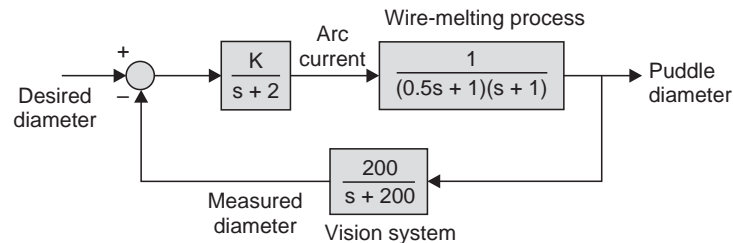



Fig. P7.3 Welder control

P7.4 The block diagram of a remotely guided vehicle on Mars is shown in Fig. P7.4. Both the front and back wheels of the vehicle can be steered. Compute (*a*) the value of K required for stability and (*b*) the value of K when one root of the characteristic equation is to be placed at $s = -4$. (*c*) Also find the response of the system subjected to a step command for the gain selected in part (*b*).

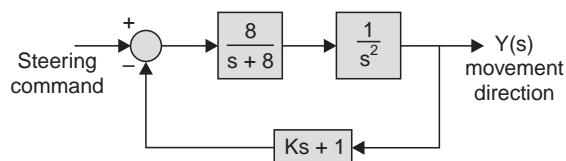



Fig. P7.4 Guided vehicle control

CHAPTER

8

Design of Compensators

8.1 INTRODUCTION

We mentioned in Section 1.7 of Chapter 1 that one of the important tasks of a control engineer is to modify the structure of the control system if the analysis of the basic system reveals some sort of shortcoming in meeting one or more of the performance specifications in the steady state and/or transient state.

In redesigning a control system to alter the system response, an additional component is inserted within the structure of the feedback system. It may be accomplished by introducing additional feedback or by incorporating additional components to modify the behavior of the overall system. The additional components introduced to compensate the deficiency in performance are referred to as compensators whereas additional feedback employs output or state feedback.

So, the design of a control system is concerned with the modification of the structure of the basic system and the choice of suitable components and parameters in the forward and feedback path for compensating the deficiency in performance of the basic system.

The physical nature of the components used as *compensator* may be electric, mechanical, electromechanical, hydraulic, pneumatic, or some other type of device or network depending on the physical nature of the system. But whenever feasible, the device is chosen to produce electrical output for ease of signal processing.

It is normally assumed that the plant or process itself cannot be readily changed, even though the direction of design improvements may be the direct fallout of the detailed analysis of the system behavior. For example, the aerodynamic design of an aircraft may be undertaken for improving maneuverability of a fighter plane by modifying the wing and control surface areas. However, the control engineer should reconcile to the fact that plant or process is unalterable and the augmentation of the basic system with some sort of compensators, including additional feedbacks, if feasible, are the options left to improve the system performance.

It has to be mentioned at this stage that even with the introduction of compensators, not all the desired performance specifications, which are sometimes conflicting and demanding, can be satisfied simultaneously and a compromise solution has to be accepted.

8.2 APPROACHES TO SYSTEM DESIGN

We have noted that the performance of a control system can be described in terms of the time-domain as well as in frequency-domain parameters. The time-domain parameters include the *peak time* t_p*, maximum overshoot, and settling-time* for a step input, the *maximum allowable steady-state error* for typical test inputs and disturbance inputs. It has been noticed from the

time-domain analysis that the system behavior is associated with the location of the roots of the characteristic equation of the closed loop system. So, the location of the poles and zeros of closed loop system may be specified for desired transient and steady state response. The root locus diagram may be used as a design tool when there is only one parameter (normally the forward path gain) to be adjusted.

However, when the root cannot be placed at the desired location with a single parameter variation, lag-lead compensators are incorporated at suitable location in the structure of the system to meet the design specifications.

Alternatively, state feedback may be used, since theoretically it is possible to place the poles of a closed loop system anywhere in the plane of the root if all the states are available for measurement and feedback. This concept is known as *pole assignment* and is elaborated in Chapter 9.

The performance measure of a feedback control system is also described in terms of frequency-domain parameters. The common frequency-domain parameters are the *peak frequency response,* M_p*, the resonant frequency* ω_r*, the bandwidth, and the phase margin and gain margin* of the closed loop system. We can add a suitable compensation network, when the basic system fails to meet the system specifications. The design of the compensating network may be undertaken in the frequency-domain using the Bode diagram, or the Nichols chart. The Bode diagram offers some definite advantage over the Nichols chart since the contribution from the poles and zeros of the compensator becomes additive in the log scale. The Nichols chart may be used for final checking the frequency response of the closed loop system.

With this end in view we present the design of compensators for the analog control systems in this chapter followed by pole assignment design in the next chapter.

8.2.1 Structure of the Compensated System

A compensator, designated with the transfer function $D_c(s)$, can be introduced in a control system in a number of alternative locations within the structure of the system. The commonly used positions for a single-loop feedback control system are indicated in Fig. 8.1 and these are referred to by their position in the system structure: (*a*) *cascade or series compensator,* (*b*) *feedback or parallel,* (*c*) *input and* (*d*) *output compensator*. The choice in a particular situation depends on many factors like signal power level, physical nature of the component, feasibility of implementation. However, the most widely used configuration is the cascade and feedback compensator and so far as design steps are concerned, their treatment are similar.

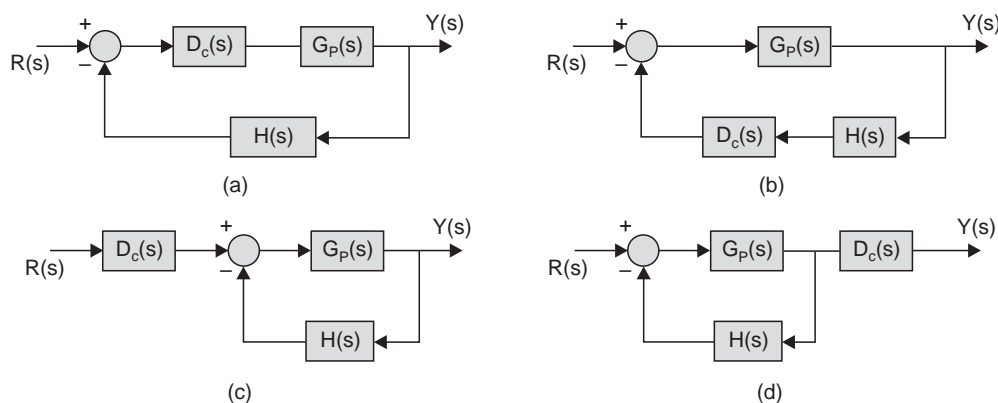



Fig. 8.1 Compensator configurations (a) series (b) feedback or parallel (c) input and (d) output compensation

8.2.2 Cascade Compensation Networks

The compensator $D_c(s)$ can be chosen to alter either the shape of the root locus or the frequency response. The transfer function of a compensator in its most general form may be represented as:

$$D_c(s) = \frac{K_c \prod_{i=1}^{q} (s + a_i)}{\prod_{j=1}^{n} (s + b_j)} \tag{8.1}$$

With reference to Fig. 8.1(a), the loop transfer function is, therefore, written as $D_c(s)G_p(s)H(s)$, where $G_p(s)$ is the plant transfer function and $H(s)$ is the feedback path transfer function.

The design problem then reduces to the choice of the poles and zeros of the compensator to provide a loop transfer function $D_c(s)G_p(s)H(s)$ with desired frequency response or desired closed loop system poles.

In order to illustrate the design steps, we consider a first order compensator with $n = q = 1$ such that it is represented as

$$\frac{E_o(s)}{E_i(s)} = D_c(s) = \frac{K_c(s + a_1)}{(s + b_1)} \tag{8.2}$$

The frequency response of the compensator is obtained as

$$\frac{E_o(j\omega)}{E_i(j\omega)} = D_c(j\omega) = \frac{K_c(j\omega + a_1)}{(j\omega + b_1)} = \frac{a_o(1 + j\alpha\omega\tau)}{1 + j\omega\tau} \tag{8.3}$$

where $\tau = 1/b_1$, $b_1 = \alpha a_1$, and $a_o = K_c/\alpha$.

(a) **Phase-Lead Compensator:** When $|a_1| < |b_1|$, the compensator is called a **phase-lead compensator** since, the phase shift of the output $E_o(j\omega)$ will lead over that of the input $E_i(j\omega)$ and has a pole-zero configuration in the s-plane, as shown in Fig. 8.2.

The frequency response of the network can be obtained by Bode diagram and is shown in Fig. 8.3. The phase angle of the network is given by

$$\phi(\omega) = \tan^{-1} \alpha\omega\tau - \tan^{-1} \omega\tau \tag{8.4}$$

From the Bode plot we find that the overall phase angle is positive. This is due to the fact that the zero occurs first on the frequency axis than the occurrence of pole.

Incidentally, the phase-lead compensation transfer function can be realized with the help of the RC components in conjunction with an operational amplifier as shown in Fig. 8.4. The transfer function of this network is given by

$$D(s) = \frac{E_o(s)}{E_i(s)} = \frac{R_2}{R_2 + Z}, \text{ where } Z = \frac{\frac{R_1}{Cs}}{R_1 + \frac{1}{Cs}} = \frac{R_1}{sCR_1 + 1}$$

or

$$D(s) = \frac{1}{\alpha}\frac{(1 + \alpha\tau s)}{(1 + \tau s)} = \frac{s + a_1}{s + b_1} \tag{8.5}$$

where

$$\tau = \frac{R_1R_2}{R_1 + R_2} C,\ \alpha = \frac{R_1 + R_2}{R_2} > 1,\ b_1 = 1/\tau \quad \text{and} \quad a_1 = 1/\alpha\tau \tag{8.6}$$

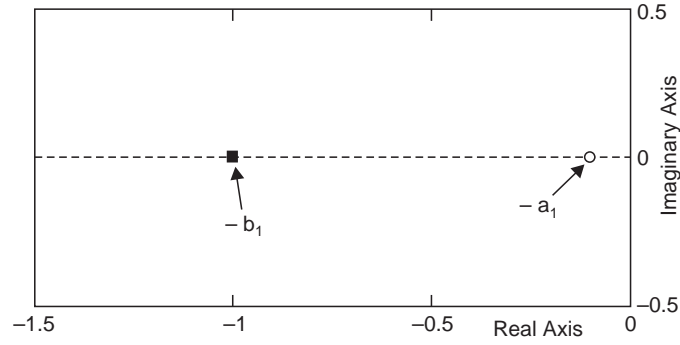



Fig. 8.2 Pole-zero location of phase lead compensator

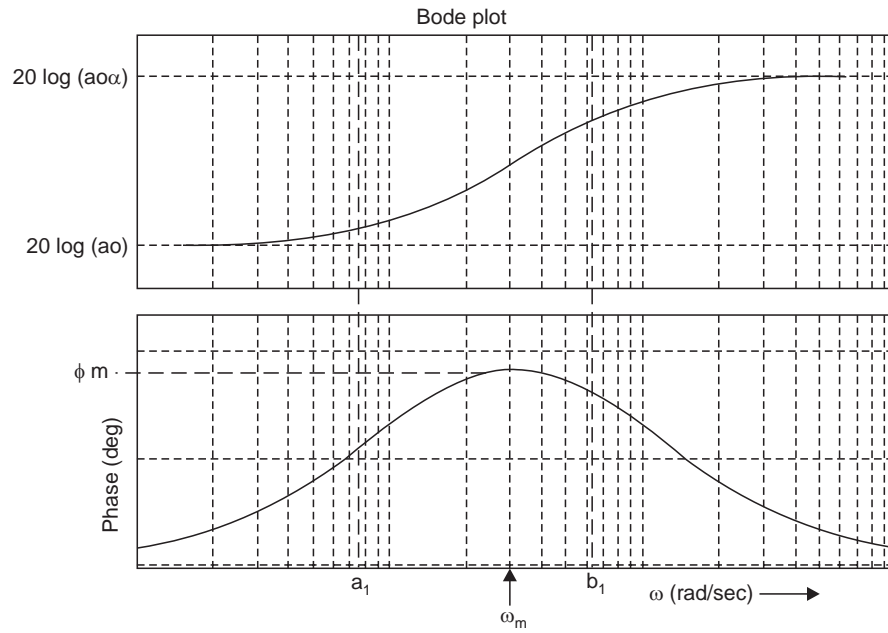



Fig. 8.3 Bode plot of phase-lead compensator

At very low frequency ($s \to 0$) the compensator introduces an attenuation of $1/\alpha$, which can be taken care of by the accompanying amplifier together with any additional gain needed to meet the steady state error specification. So the phase-lead compensator in Equation (8.3) can be readily implemented by the RC network of Fig. 8.4.

The frequency, ω_m, for which the phase-lead compensator introduces maximum phase shift can be obtained by solving the equation $\frac{d(\tan \phi)}{d\omega} = 0$, where ϕ is given by expression (8.4). This yields

$$\omega_m = \frac{1}{\tau\sqrt{\alpha}} = \sqrt{a_1 b_1} \tag{8.7}$$

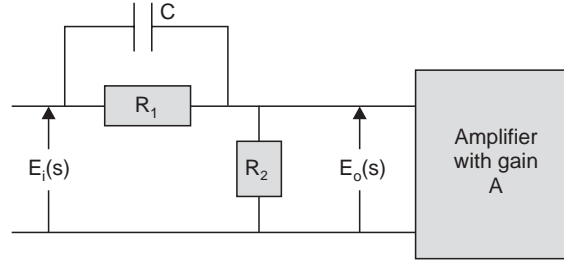



Fig. 8.4 Phase lead network

So, the phase shift introduced by phase-lead network attains a maximum value of ϕ_m at a frequency ω_m which is the geometric mean of the poles and zeros of the compensator, where ϕ_m is given by:

$$\phi_m = \tan^{-1}\left(\frac{\alpha - 1}{2\sqrt{\alpha}}\right) \tag{8.8}$$

Alternatively, α can be solved from the properties of right angled triangle for a prescribed value of ϕ_m as

$$\alpha = (1 + \sin \phi_m)/(1 - \sin \phi_m) \tag{8.9}$$

Since, α is the ratio of pole to zero, ($\alpha = b_1/a_1$), for a desired maximum phase ϕ_m, the relation (8.8) may be used to find the pole/zero ratio of the compensator. The variation of ϕ_m with α is shown in Fig. 8.5, which reveals that practically a maximum phase shift of about 70° or less is feasible from a single section RC network. If a phase shift more than 70° is needed then two phase-lead networks may be cascaded.

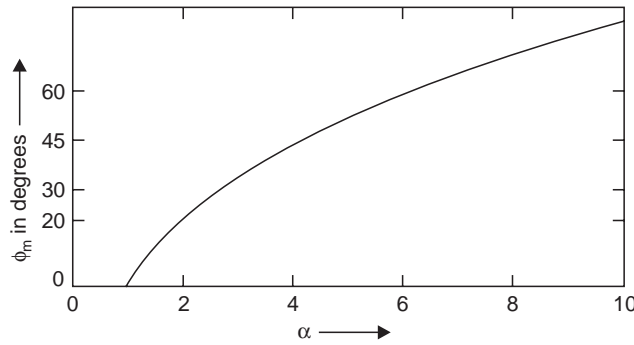



Fig. 8.5 Variation of maximum phase with pole-zero ratios

(*b*) **Phase Lag Compensator:** When $|a_1| > |b_1|$, the compensator in relation (8.2) is known as **Phase-Lag Compensator.** In this case, since, $|b_1| < |a_1|$, the pole lies closer to the origin than the zero in the *s*-plane, as shown in Fig. 8.6. The Bode plot of the phase-lag compensator is shown in Fig. 8.7. From the Bode plot in Fig. 8.7, the overall phase shift is found to be negative and the phase of the output signal will be lagging behind that of the input and hence the name Phase Lag network.

The realization of the phase-lag network using RC components is shown in Fig. 8.8.

The transfer function of the phase-lag network of Fig. 8.8 is given by

$$D(s) = \frac{E_o(s)}{E_i(s)} = \frac{R_2 + (1/Cs)}{R_1 + R_2 + (1/Cs)} = \frac{R_2Cs + 1}{(R_1 + R_2)Cs + 1}$$

$$= \frac{1+\tau s}{1+\alpha\tau s} = \frac{1}{\alpha}\frac{(s+a_1)}{(s+b_1)} \tag{8.10}$$

where $\tau = R_2C$, $\alpha = (R_1 + R_2)/R_2 > 1$, $a_1 = 1/\tau$ and $b_1 = 1/(\alpha\tau)$.

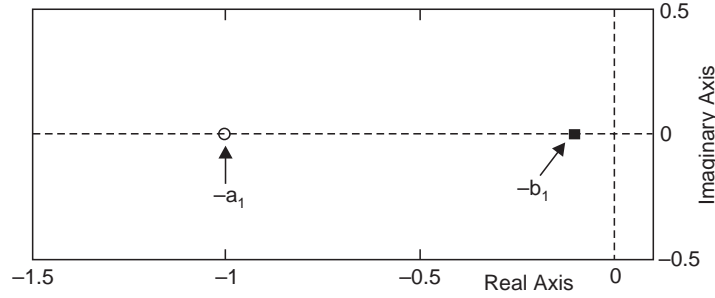



Fig. 8.6 Pole-zero location of phase-lag compensator

It can be shown that for the phase-lag network also, the frequency at which maximum phase shift occurs is the geometric mean of the pole and zero. So, it is given by

$$\omega_m = \sqrt{a_1 b_1}$$

and the magnitude of ϕ_m is numerically the same as in Equation (8.8), but negative in sign.

The phase-lead network is intended to *introduce the phase-lead angle* to meet the phase margin specification whereas the phase-lag network is intended *to provide attenuation* that will increase the steady-error constant. Its phase lag is only incidental and *not intended and is kept as small as possible* ($\leq |\ 5°\ |$). The phase-lead network modifies the root locus to move it to the desired root locations and affects the transient response. The phase-lag network is not intended to modify the root locus and the transient response but *to influence the steady state response*.

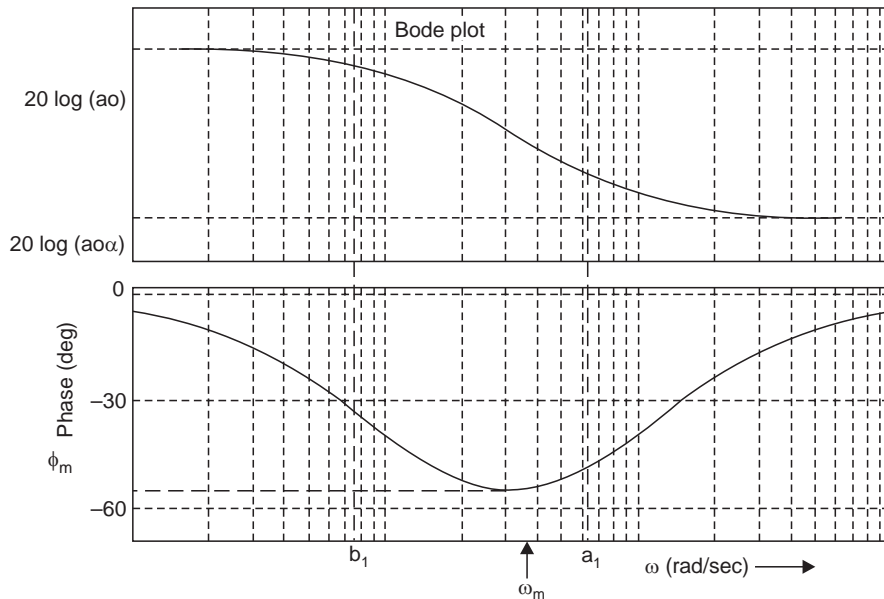



Fig. 8.7 Bode plot of phase-lag compensator

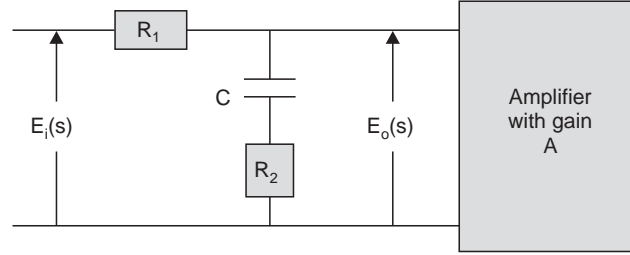



Fig. 8.8 Phase-lag network

8.2.3 Design Concept for Lag or Lead Compensator in Frequency-Domain

We consider the analog system in Fig. 8.9(*a*) in *s* domain with unity feedback. The results that are developed below are also applicable to cases where the plant transfer function is $G_p(s)$ and feedback is $H(s)$. This can be done just by replacing $G(j\omega)$ in the equations presented below by $G_p(j\omega)H(j\omega)$. The Nyquist plot for $G(j\omega)$ is shown in Fig. 8.9(*b*). We shall design the compensator to meet the specification of phase margin ϕ_m by reshaping the Nyquist plot around the point where it crosses the unity circle as shown in Fig. 8.9(*b*) by the dotted curve. This can be accomplished by any one of the following approaches or their combination:

(*i*) *by introducing phase lead around the frequency* $\omega = \omega_1$ *, where the Nyquist plot crosses the unity circle and is realised by using phase-lead network.*

(*ii*) *by adding additional attenuation around the frequency* $\omega = \omega_1$*, and is realised by phase-lag network.*

The characteristic equation of the analog system incorporating the compensator is given by

$$1 + D_c(s)G(s) = 0$$

With $s = j\omega$, the above equation can be rearranged as:

$$D_c(j\omega) = -\,1/G(j\omega) \tag{8.11}$$

If ω_1 is the frequency at which the Nyquist plot of the compensated system crosses the unity circle [vide Fig. 8.9(*b*)], the phase angle θ of the compensator must satisfy, at $\omega = \omega_1$, the condition,

$$\theta = \angle D_c(j\omega_1) = -\,180° + \phi_m - \angle G(j\omega_1) \tag{8.12}$$

where ϕ_m is the Phase Margin.

The Equation (8.11) at $\omega = \omega_1$, can, therefore, be written as $D_c(j\omega_1) = r\angle\theta$ (8.13)

where $r = \left|\dfrac{1}{G(j\omega_1)}\right|$ and θ is the phase angle of the compensator $D_c(j\omega)$ at $\omega = \omega_1$.

Now from relation (8.3), we have $D_c(j\omega) = \dfrac{\alpha_o + j\alpha_1\omega}{1 + j\beta_1\omega}$ (8.14)

where $\alpha_0 = \dfrac{K_c a_1}{b_1}$, $\alpha_1 = \dfrac{K_c}{b_1}$ and $\beta_1 = \dfrac{1}{b_1}$

From relations (8.13) and (8.14) we have,

$$D_c(j\omega_1) = \frac{\alpha_0 + j\alpha_1\omega_1}{1 + j\beta_1\omega_1} = \frac{1\angle\theta}{|G(j\omega_1)|}$$

$$\Rightarrow \qquad \frac{\alpha_0 + j\alpha_1\omega_1}{1 + j\beta_1\omega_1} = \frac{\cos\theta + j\sin\theta}{|G(j\omega_1)|} \tag{8.15}$$

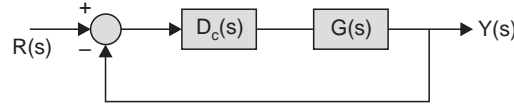
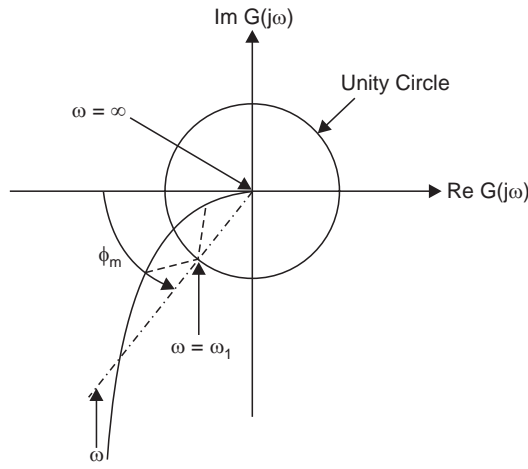
or, $\alpha_o |\, G(j\omega_1) \,| + j\alpha_1\omega_1 \,|\, G(j\omega_1) \,| = \cos\theta - \beta_1\omega_1 \sin\theta + j(\sin\theta + \beta_1\omega_1\cos\theta)$

(a) Series compensator

(b) Nyquist plot of G (jω)

Fig 8.9 (a) Control system with transfer function G(*s*) with series compensator $D_c(s)$
(b) Nyquist plot of G(*j*ω) and its modification by $D_c(j\omega)$

Equating real parts, we get

$$\cos\theta - \beta_1\omega_1 \sin\theta = \alpha_o \,|\, G(j\omega_1) \,|$$

$$\Rightarrow \qquad \beta_1 = \frac{\cos\theta - \alpha_o |G(j\omega_1)|}{\omega_1 \sin\theta} \tag{8.16}$$

Similarly, equating imaginary parts and using Equation (8.16) and simplification, we get

$$\alpha_1 = \frac{1 - \alpha_0 |G(j\omega_1)| \cos\theta}{\omega_1 |G(j\omega_1)| \sin\theta} \tag{8.17}$$

It should be noted that (8.16) and (8.17) may be utilized for designing both types of compensators, phase-lag as well as phase-lead, since no specific assumption was made about any particular network in the development of these equations.

This design procedure requires that the compensator dc gain α_o and the system phase-margin frequency ω_1 be chosen from the frequency response of the loop gain. Then the Equations (8.16) and (8.17) determine the compensator parameters α_1 and β_1. The compensator dc gain α_o is usually merged with the forward path gain K of the system, which is determined by steady state specifications for the control system. The frequency ω_1 must be chosen satisfying the two design steps given below for the Phase Lead and Phase Lag Network.

8.2.4 Design Steps for Lag Compensator

Let θ be the phase angle of $D_c(s)$ at the gain crossover point for the Lag Network, which is negative.

Step 1: Find ω_1 from frequency response of G(jω), such that

$$\angle G(j\omega_1) \geq -180° + \phi_m - \theta, \; -5° \geq \theta > -10°$$

For a stable compensator β_1 should be positive and, since θ and sin θ are both negative, the next design step can be written from relation (8.16) as:

Step 2: $\alpha_o \,|\, G(j\omega_1) \,| > \cos\theta$

From the point of view of realization of the lag compensator by RC network of Fig. 8.8, the value of $|\, G(j\omega_1) \,|$ in step 2 should preferably be less than 10. If the value of $|\, G(j\omega_1) \,|$ becomes more than 10, it implies that more than one section of lag compensator or lag lead compensator should be used (vide Example 8.4).

8.2.5 Design Steps for Lead Compensator

For Lead Network θ is positive.

Step 1: Find ω_1 from frequency response of G(jω), such that

$$\angle G(j\omega_1) \leq -180° + \phi_m - \theta, \text{ for } 70° \geq \theta > \phi_m + 5° - \phi_{mu},$$

where, ϕ_{mu} is the phase margin of the uncompensated system, and 5° is added as a safety measure.

For a stable compensator β_1 should be positive. Hence for a positive θ and sin θ we must have

Step 2: $\cos\theta > \alpha_o \,|\, G(j\omega_1) \,|$

8.2.6 Design Examples

We consider here some examples to illustrate the above design steps that will provide the specified phase margin in the control system, provided the designed system is stable. That is, the procedure will set the gain and phase of the open-loop transfer function to specified values at a given frequency, and we choose the specified gain to be 0 db and the specified phase to be $(180° + \phi_m)$, where ϕ_m is the desired phase margin. The procedure, therefore, does not ensure the stability of the closed loop system, which should be checked separately by finding the gain margin.

Example 8.1 Design of Phase-Lead Network

We consider the system in Fig. 8.9(*a*) with $G(s) = \dfrac{2.5}{s(0.2s+1)(0.5s+1)}$ and $H(s) = 1$.

We are interested to design a compensator to provide a phase margin ϕ_m of 45°. The frequency response of plant transfer function is summarized in Table 8.1. Since, the phase margin of the uncompensated system ϕ_{mu}, is found to be 29.1°, at $\omega_p = 1.77$ rad/sec, the additional phase lead to be introduced by the compensator, $\theta \geq 21° \approx \phi_m + 5° - \phi_{mu}$. So, from design step 1 of phase-lead compensator, a frequency ω_1 is to be found for which

$$\angle G(j\omega_1) \leq -159° = -180° + 21°.$$

Table 8.1 Frequency response of plant $G_p(s) = \dfrac{25}{s(s+2)(s+5)}$

Angular frequency ω	*Amplitude* | $G(j\omega)$ |	*Phase angle*	*Angular frequency* ω	*Amplitude* | $G(j\omega)$ |	*Phase angle*
0.1000	24.9638	– 94.0082	2.6000	0.5201	– 169.9058
0.2000	12.4280	– 98.0012	2.7000	0.4849	– 171.8402
0.3000	8.2263	– 101.9644	2.8000	0.4528	– 173.7111
0.4000	6.1091	– 105.8839	2.9000	0.4234	– 175.5214
0.5000	4.8266	– 109.7468	3.0000	0.3964	– 177.2737
0.6000	3.9625	– 113.5420	3.1000	0.3716	– 178.9704
0.7000	3.3384	– 117.2597	3.2000	0.3488	– 180.6139
0.8000	2.8650	– 120.8917	3.3000	0.3277	– 182.2064
0.9000	2.4930	– 124.4317	3.4000	0.3083	– 183.7502
1.0000	2.1926	– 127.8750	3.5000	0.2903	– 185.2471
1.1000	1.9449	– 131.2182	3.6000	0.2737	– 186.6993
1.2000	1.7371	– 134.4595	3.7000	0.2583	– 188.1084
1.3000	1.5605	– 137.5981	3.8000	0.2440	– 189.4763
1.4000	1.4087	– 140.6343	3.9000	0.2306	– 190.8045
1.5000	1.2771	– 143.5691	4.0000	0.2183	– 192.0948
1.6000	1.1621	– 146.4045	4.1000	0.2067	– 193.3484
1.7000	1.0609	– 149.1426	4.2000	0.1960	– 194.5669
1.8000	0.9713	– 151.7861	4.3000	0.1859	– 195.7516
1.9000	0.8917	– 154.3380	4.4000	0.1765	– 196.9038
2.0000	0.8207	– 156.8014	4.5000	0.1677	– 198.0247
2.1000	0.7570	– 159.1796	4.6000	0.1595	– 199.1155
2.2000	0.6997	– 161.4758	4.7000	0.1518	– 200.1772
2.3000	0.6480	– 163.6933	4.8000	0.1445	– 201.2110
2.4000	0.6012	– 165.8354	4.9000	0.1377	– 202.2178
2.5000	0.5587	– 167.9052	5.0000	0.1313	– 203.1986

From the table, it is found that any frequency $\omega_1 > 2.1000$ radian will satisfy the condition of design step 1. Now choosing $\omega_1 = 2.2000$, we note that $\angle G(j\omega_1) = -161.4758° \le -159°$ and $| G(j\omega_1) | = 0.6997$. Thus we find $\theta = 26.4758° = -180° + 45° + 161.4758°$ and cos 26.4758° = 0.8951 such that with $\alpha_o = 1$, the condition of step 2 is satisfied as 0.8951 > 0.6997. Using Equations (8.16) and (8.17), the compensator parameters α_1 and β_1 are found to be: $\alpha_1 = 0.5445$, $\beta_1 = 0.1992$ respectively. Hence the compensator in s domain is given by

$$D_c(s) = \frac{\alpha_0 + \alpha_1 s}{1 + \beta_1 s} = \frac{1 + 0.5445s}{1 + 0.1992s} = \frac{2.7329(s + 1.8365)}{s + 5.0189}$$

The above compensator may be realized by the RC network of Fig. 8.4 with C = 1 microfarad, R_1 = 108.49 K ohm, R_2 = 62.610 K ohm and amplifier gain A = 2.7329. The important frequency-domain parameters for the lead compensated system are found to be GM = 11.9 dB, PM = 45°, phase crossover frequency ω_g = 5.15 rad/sec, gain crossover frequency ω_p = 2.2 rad/sec and BW = 2.83 rad/sec and whereas that of the uncompensated system are given by GM = 8.94 dB, PM = 29.1°, ω_g = 3.16 rad/sec, ω_p = 1.77 rad/sec and BW = 2.18 rad/sec.

The Bode plot as well as step response of the uncompensated, lead compensated and lag compensated system (vide Example 8.2 below) are shown in Figs. 8.10 and 8.11 respectively.

Example 8.2 **Design of Phase-Lag Network**

Consider again the system of Example 8.1, whose frequency response is given in Table 8.1. A phase margin $\phi_m = 45°$ is to be achieved, and a unity-gain phase-lag compensator will be employed.

Let $\theta = -5°$, from step 1, $\angle G(j\omega_{w1}) \ge -180° + 45° - (-5°) = -130°$; we must choose a frequency ω_1 such that $\angle G(j\omega_1) \ge -130°$. From Table 1 it is found that any frequency $\omega_1 \le 1.100$ radian will satisfy this constraint. Now the second constraint, $| G(j\omega_1) | > \cos 5° = 0.9962$ must also be satisfied. We rather arbitrarily choose $\omega_1 \cong 1.0000$ to satisfy these two constraints, since $| G(j\omega_1) | = 2.1926$ such that constraint 2, (2.1926 > cos 5 = 0.9962) is also satisfied at $\omega_1 \cong 1.0000$. Hence, from (8.17),

$$\alpha_1 = \{1 - (1)(2.1926) \cos (-5°)\}/\{(1)\ (2.1926) \sin (-5°)\} = 6.1971$$

$$\beta_1 = \{\cos (-5°) - (1)\ (2.1926)\}/\{(1.0) \sin (-5°)\} = 13.7272$$

and thus, from (8.3), we have $D(s) = \dfrac{0.4514(s + 0.1614)}{(s + 0.0728)}$

The frequency-domain parameters of the lag compensated system are found to be

GM = 15.3 dB, PM = 47.1°, ω_g = 3.06, ω_p = 1 rad/sec and BW = 1.3 rad/sec

The important parameters for step response of the system with unity feedback, along with the frequency-domain parameters, are shown in Table 8.2.

From the table we find that the overshoot of the lag and lead compensated system are almost equal. However, the response of the lead compensated system is much faster (smaller rise time and settling time) than that of the uncompensated and lag compensated system.

Table 8.2

Parameter	*Un-compensated system*	*Lead-compensated system*	*Lag-compensated system*
Peak response, M_p	1.44	1.24	1.24
First peak	1.78 sec	1.31 sec	2.79 sec
% Overshoot	43.5	24	24
Rise time, t_r	0.667 sec	0.538 sec	1.14 sec
Settling Time, t_s	7.12 sec	2.95 sec	9.5 sec
PM	29.1°	45°	47.1°
GM	8.94 dB	11.9 dB	15.3 dB
Gain cross-over frequency, ω_p	1.77 rad/sec	2.2 rad/sec	1 rad/sec
Phase cross-over frequency ω_g	3.16 rad/sec	5.15 rad/sec	3.06 rad/sec
BW	2.18 rad/sec	2.83 rad/sec	1.3 rad/sec

The phase-lag compensator of Fig. 8.8 may be realized with C = 1.0 micro farad, R_1 = 7.5301 M ohm and R_2 = 6.1971 M ohm.

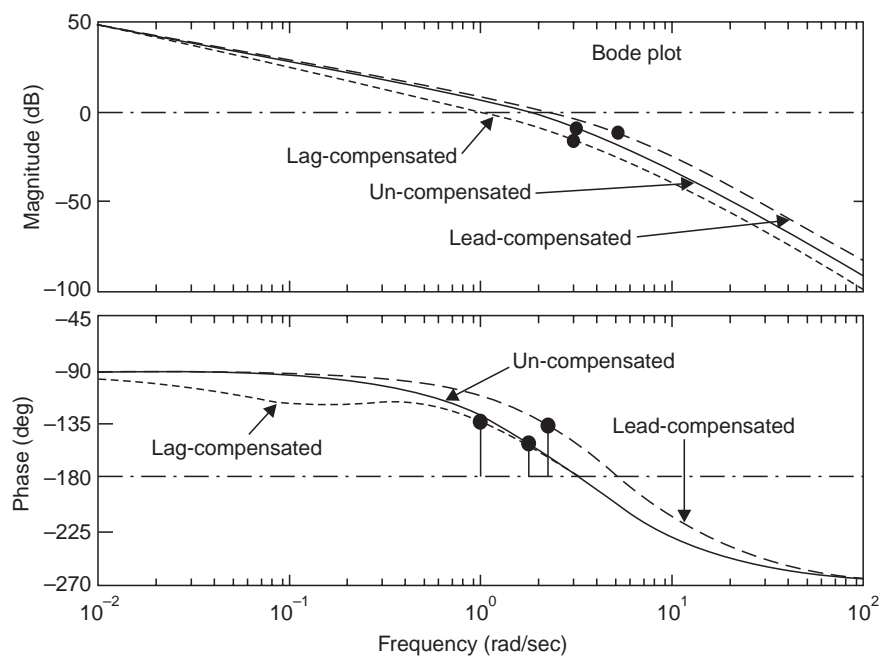



Fig. 8.10 Bode plot of uncompensated, lead-compensated and lag-compensated system

Chapter 8

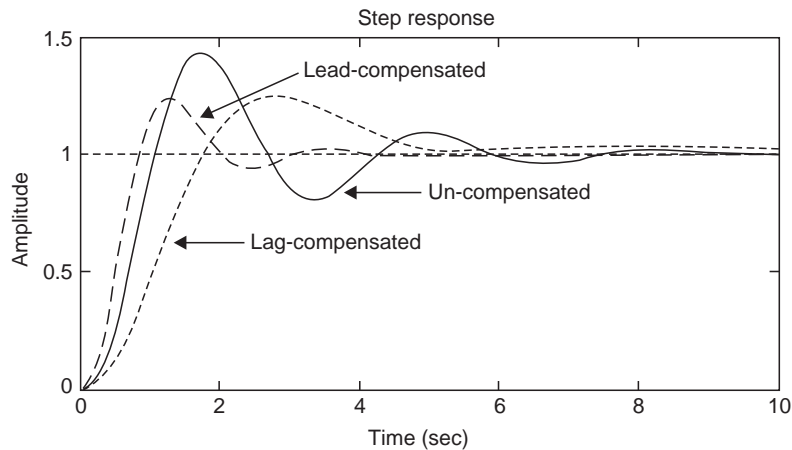



Fig. 8.11 Step response of uncompensated, lead-compensated and lag-compensated system of Example 8.1

A choice of frequency too far below from ω_1 may result in a compensator with poles and zeros that is difficult to realize using practical RC values. For example, if we take $\omega_1 = 0.3$ for which $G(j\omega_1) = 8.2263$ results in a G(*s*) given by:

$$D(s) = \frac{0.1210(s + 0.0299)}{(s + 0.0036)}$$

With a C = 1.0 micro farad, the resistances are found to be, R_1 = 243.07 M ohm and R_2 = 33.451 M ohm, which are rather on the higher side.

Example 8.3 Let us consider a second order plant given by $G_p(s) = \dfrac{K}{s(s+4)}$.

For a unity ramp input, the desired steady state error of the closed loop system with unity feedback should be less than 0.05 and the damping ratio, δ of the closed loop system is 0.6.

Now since $e_{ss} \le 0.05$, by using relation (4.33) of Chapter 4, we find $K \ge 80$. We take K = 85.

Table 8.3 shows the frequency response of $G_p(j\omega)$ with K = 85.

Table 8.3 Frequency Response of $G_p(s) = \dfrac{85}{s(s+4)}$

Angular frequency ω	*Amplitude* | *G(j*ω*)* |	*Phase angle*	*Angular frequency* ω	*Amplitude* | *G(j*ω*)* |	*Phase angle*
1	20.6155	– 104.0362	26	0.1243	– 171.2538
2	9.5033	– 116.5651	27	0.1153	– 171.5730
3	5.6667	– 126.8699	28	0.1073	– 171.8699
4	3.7565	– 135.0000	29	0.1001	– 172.1467
5	2.6550	– 141.3402	30	0.0936	– 172.4054
6	1.9646	– 146.3099	31	0.0877	– 172.6476
7	1.5061	– 150.2551	32	0.0824	– 172.8750

(Contd.)...

8	1.1879	– 153.4349	33	0.0775	– 173.0888
9	0.9589	– 156.0375	34	0.0730	– 173.2902
10	0.7892	– 158.1986	35	0.0689	– 173.4802
11	0.6602	– 160.0169	36	0.0652	– 173.6598
12	0.5600	– 161.5651	37	0.0617	– 173.8298
13	0.4807	– 162.8973	38	0.0585	– 173.9910
14	0.4170	– 164.0546	39	0.0556	– 174.1440
15	0.3650	– 165.0686	40	0.0529	– 174.2894
16	0.3221	– 165.9638	41	0.0503	– 174.4278
17	0.2863	– 166.7595	42	0.0480	– 174.5597
18	0.2561	– 167.4712	43	0.0458	– 174.6855
19	0.2304	– 168.1113	44	0.0437	– 174.8056
20	0.2084	– 168.6901	45	0.0418	– 174.9204
21	0.1893	– 169.2157	46	0.0400	– 175.0303
22	0.1728	– 169.6952	47	0.0383	– 175.1355
23	0.1583	– 170.1342	48	0.0368	– 175.2364
24	0.1456	– 170.5377	49	0.0353	– 175.3331
25	0.1343	– 170.9097	50	0.0339	– 175.4261

(A) Phase-lead Network

For $\delta = 0.6$, the ϕ_m may be approximated (vide Section 6.4.1) as $100\delta = 60°$. From Table 8.3, we find the phase margin of the uncompensated system, ϕ_{mu} is 24°. So a phase-lead of about $(60° + 5° - 24°) = 41°$ is to be introduced by the compensator. So we choose ω_1 such that $G(j\omega_1) \le -161° = -180° + 60° - 41°$. From Table 8.3, we see that any frequency $\omega_1 \ge 12$ rad/sec will satisfy the step 1 for phase-lead network. We take $\omega_1 = 12$ for which $G(j\omega_1) = 0.5600$. Following the procedure of Example 8.1 we get phase-lead compensator:

$$D_c(s) = \frac{3.1247(s + 6.6561)}{(s + 20.7979)}$$

This compensator can be realized with the help of network of Fig. 8.4 with $R_1 = 3.0804$ K ohm, $R_2 = 0.6826$ ohm and C = 1 micro farad and amplifier gain of A = 3.1247.

The important frequency-domain parameters of the closed loop system are presented in Table 8.4.

(B) Phase-lag Network

Taking $\theta = -5°$ for the phase-lag network, the condition of step 1 becomes,

$$G(j\omega_1) \ge -180° + \phi_m - \theta = -115°.$$

We see from Table 8.3 that any frequency $\omega_1 \le 2$ rad/sec will satisfy the above condition. We take $\omega_1 = 2$ rad/sec and the corresponding value of $G(j\omega_1) = 9.5033$, is found to satisfy the condition in step (9.5033 > 0.9962 = cos 5). This results in the lag compensator

$$D_c(s) = \frac{0.1047(s + 0.1956)}{(s + 0.0205)}$$

The compensator can be realized by the RC network of Fig. 8.8, with R_1 = 43.693 M ohm, R_2 = 5.1114 M ohm, C = 1 micro farad and an amplifier gain of A = 0.1047.

The Bode plot and step response of the system are shown in Figs. 8.12 and 8.13 respectively and the important time and frequency-domain parameters are shown in Table 8.4.

Table 8.4

Parameter	*Uncompensated system*	*Lead-compensated system*	*Lag-compensated system*
PM	24.5°	60°	58.4°
GM	∞	∞	∞
ω_p	8.8 rad/sec	12 rad/sec	2 rad/sec
BW	10.6 rad/sec	15.85 rad/sec	2.63 rad/sec
Peak response, M_p	1.5	1.13	1.14
Peak time	0.348 sec	0.25 sec	1.44 sec
% Overshoot	49.7	12.7	13.9
Rise time, t_r	0.133 sec	0.106 sec	0.633 sec
Settling time, t_s	1.84 sec	0.488 sec	7.48 sec

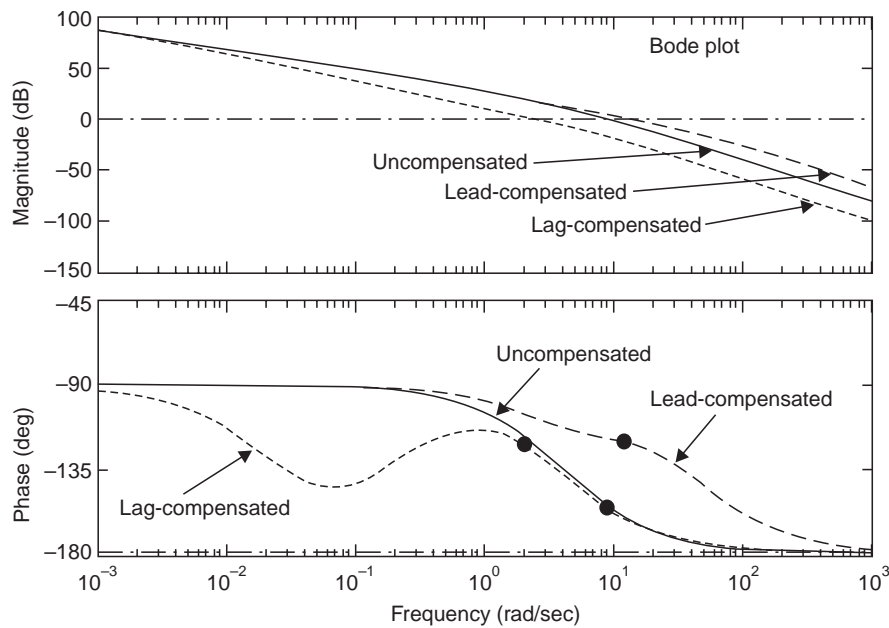



Fig. 8.12 Bode plot of uncompensated, lead-compensated and lag-compensated system in Example 8.3

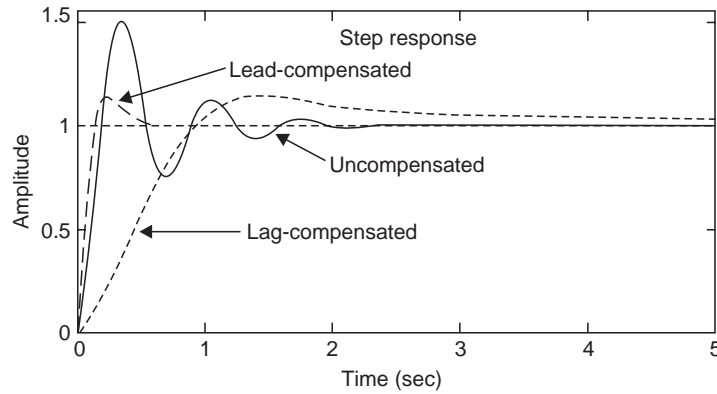



Fig. 8.13 Step response of uncompensated, lead-compensated and lag-compensated system in Example 8.3

The time-domain and frequency-domain performance of the lead and lag compensated systems reveals that the lead compensator increases the bandwidth and improves transient response and reduces the overshoot. However, because of the increased bandwidth, the high frequency noise signals will also be amplified, which is not desirable. The lag compensators, on the other hand, decreases the bandwidth making the response sluggish, attenuates high frequency signals and increases low frequency amplitude and improves steady state performance. So, a combination of lag-lead compensator is expected to combine the benefits of steady state performance of lag network with the improved transient response of the phase-lead compensator.

The transfer function of Proportional Derivative (PD) controller is given by

$$G_c(s) = K_p + K_d\, s = K_p(1 + sT_d) \tag{8.18}$$

So, it is a special form of lead compensator, where the pole located at $-b_1$ is absent. The constant K_p is normally set to meet the steady state error specification and the corner frequency $1/T_d$ is chosen to introduce the phase lead at the gain cross over frequency to meet the specification on phase margin. Because of the absence of any pole, the high frequency response beyond $\omega > 1/T_d$ will continue to increase for the PD controller and phase-lead compensator is preferred to PD controller on this account. The realization of the PD controller by passive RC network is not feasible because of the presence of a simple zero unaccompanied by a pole in its transfer function. However, the PD controller may be realized by active electronic circuits, Hydraulic and Pneumatic components.

Let us now examine the transfer function of PI controller given by

$$G_c(s) = K_p + \frac{K_I}{s} = \frac{K_p\left(s + \frac{K_I}{K_p}\right)}{s} \tag{8.19}$$

It is observed that the PI controller is a special form of phase-lag compensator whose pole is located at the origin of the s-plane. The presence of the pole at the origin makes the gain at zero frequency infinite, improving the steady state performance. Its presence will degrade the relative stability of the closed loop system, even may result in an unstable system. Apart from this, because of the pure integration term in relation (8.19), the PI compensator cannot be realized by passive RC circuits of Fig. 8.8.

Lag-lead Compensator

It is apparent that, both the steady state performance and the transient response may be taken care of by using a lag-lead compensator. For realization of lag-lead compensators, separate lag and lead circuits of Figs. 8.4 and 8.8 need not be used in cascade, even though there is no harm in doing so when they are isolated by amplifiers. The RC circuits like the one in Fig. 8.14 may be used instead. Mechanical lag-lead compensator may be realized using Dashpots and Springs. Hydraulic and Pneumatic lag-lead compensators may also be realized with bellows and resistors to fluid flow. The transfer function of the lag-lead compensator in Fig. 8.14 can be shown to be:

$$D_c(s) = \frac{(sT_1 + 1)(sT_2 + 1)}{\left(s\frac{T_1}{\alpha} + 1\right)(s\alpha T_1 + 1)} = \frac{(s + a_1)(s + a_2)}{(s + \alpha a_1)\left(s + \frac{a_2}{\alpha}\right)} \tag{8.20}$$

where $T_1 = R_1C_1 = 1/a_1$, $T_2 = R_2C_2 = 1/a_2$, $R_1C_1 + R_2C_2 + R_1C_2 = \frac{T_1}{\alpha} + \alpha T_2$, $\alpha > 1$

We shall illustrate the design of lag-lead compensator through an example.

Example 8.4 For a system with plant transfer function $\frac{K}{s(s+1)(s+5)}$ and unity feedback, it is desired that the velocity error constant be 10 sec^{-1}, phase margin of 50° and gain margin $\geq$ 10 db.

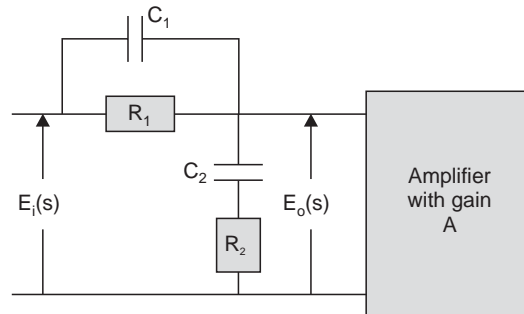



Fig. 8.14 Realization of lag-lead compensator with RC network

Since, $K_v = \lim_{s \to 0} sG(s) = \frac{K}{1 \times 5} = 0.2K$. Thus, K equals 50 for $K_v = 10$. The frequency response of the plant transfer function with K = 50 is shown in Table 8.5. From the frequency response, the phase margin and gain margin are found to be

$$\phi_{mu} = -\,10.5°, \quad GM = -\,4.44$$

So, in order to meet the steady state specification, the closed loop system with unity feedback is driven to instability. We first design a phase-lag compensator to provide the necessary attenuation.

In order to satisfy the condition in the first step of design for phase lag compensator, we must find ω_1 such that $G(j\omega_1) \geq -\,180 + \phi_m - (-\,5°) = -\,135°$. From Table 8.5, we observe that any $\omega_1 \leq 0.8$ rad / sec will meet the requirement. If we take $\omega_1 = 0.6$ with $G(j\omega_1) = 14.1897$, it will also satisfy the constraint of the second design step of the phase-lag compensator.

However, as pointed out earlier, there will be some problem in realizing the compensator, since $\alpha > 10$. So, let us take $\omega_1 = 1$ rad/sec for which $G(j\omega_1) = 6.9338$. The resulting compensator is given by

$$D_{c1}(s) = \frac{0.1435(s + 0.1023)}{(s + 0.0147)} \tag{8.21}$$

Now, the forward path transfer function is written as series combination of $D_{c1}(s)$ and $G_p(s)$ as

$$G(s) = G_p(s)\, D_{c1}(s) = \frac{7.1750(s + 0.1023)}{s(s + 1)(s + 5)(s + 0.0147)}$$

The frequency response of this cascaded transfer function is shown in Table 8.6. We note from Table 8.6 that the phase margin of the phase lag compensated system is 29°, which is the new ϕ_{mu} for the design of the phase-lead compensator. So, the required phase-lead θ must satisfy the constraint: $\theta \geq 21°$.

Consequently, we must find ω_1 such that $G(j\omega_1) \leq -180 + 50 - 29° = -159°$. From Table 8.6, we see that $\omega_1 = 1.4$ rad/sec with $\angle G(j\omega_1) = -163.6822°$ satisfies this constraint. Therefore, the exact value of θ is computed as

$$\theta = -180° + 50° + 163.6822° = 33.6822°$$

Now with this value θ and $G(j\omega_1) = 0.5752$, the phase-lead compensator is computed using relations (8.16) and (8.17) as

$$D_{c2}(s) = \frac{3.5279(s + 0.8566)}{(s + 3.0220)} \tag{8.22}$$

The relevant parameters in frequency-domain response and time-domain step response are shown in Table 8.7. The designed system is found to meet all the prescribed specifications. Fig. 8.15 shows the root locus of the lag-lead compensated system.

Table 8.5 Frequency response of $G_p(s) = \dfrac{50}{s(s + 1)(s + 5)}$

Angular frequency ω	*Amplitude* $\lvert G(j\omega) \rvert$	*Phase angle*	*Angular frequency* ω	*Amplitude* $\lvert G(j\omega) \rvert$	*Phase angle*
0.2000	48.9899	– 103.6005	5.2000	0.2517	– 215.2378
0.4000	23.1380	– 116.3753	5.4000	0.2291	– 216.7111
0.6000	14.1897	– 127.8065	5.6000	0.2091	– 218.1150
0.8000	9.6383	– 137.7501	5.8000	0.1913	– 219.4540
1.0000	6.9338	– 146.3099	6.0000	0.1754	– 220.7321
1.2000	5.1876	– 153.6902	6.2000	0.1612	– 221.9532
1.4000	3.9979	– 160.1046	6.4000	0.1485	– 223.1206
1.6000	3.1549	– 165.7393	6.6000	0.1371	– 224.2377
1.8000	2.5385	– 170.7443	6.8000	0.1267	– 225.3073
2.0000	2.0761	– 175.2364	7.0000	0.1174	– 226.3322
2.2000	1.7216	– 179.3055	7.2000	0.1090	– 227.3150
2.4000	1.4447	– 183.0211	7.4000	0.1013	– 228.2580
2.6000	1.2250	– 186.4369	7.6000	0.0943	– 229.1634

(Contd.)...

2.8000	1.0481	– 189.5950	7.8000	0.0880	– 230.0333
3.0000	0.9039	– 192.5288	8.0000	0.0822	– 230.8696
3.2000	0.7851	– 195.2652	8.2000	0.0769	– 231.6740
3.4000	0.6863	– 197.8262	8.4000	0.0720	– 232.4483
3.6000	0.6033	– 200.2298	8.6000	0.0675	– 233.1940
3.8000	0.5332	– 202.4913	8.8000	0.0634	– 233.9125
4.0000	0.4735	– 204.6236	9.0000	0.0596	– 234.6052
4.2000	0.4223	– 206.6378	9.2000	0.0561	– 235.2734
4.4000	0.3781	– 208.5435	9.4000	0.0528	– 235.9184
4.6000	0.3399	– 210.3493	9.6000	0.0499	– 236.5411
4.8000	0.3065	– 212.0626	9.8000	0.0471	– 237.1428
5.0000	0.2774	– 213.6901	10.0000	0.0445	– 237.7244

Table 8.6 Frequency response of $G(s) = \dfrac{7.175(s+0.1023)}{s(s+1)(s+5)(s+0.0147)}$

Angular frequency ω	*Amplitude* | $G(j\omega)$ |	*Phase angle*	*Angular frequency* ω	*Amplitude* | $G(j\omega)$ |	*Phase angle*
0.2000	7.8751	– 126.4866	5.2000	0.0361	– 216.2028
0.4000	3.4249	– 128.6166	5.4000	0.0329	– 217.6405
0.6000	2.0650	– 136.0790	5.6000	0.0300	– 219.0112
0.8000	1.3941	– 143.9845	5.8000	0.0275	– 220.3192
1.0000	1.0001	– 151.3088	6.0000	0.0252	– 221.5685
1.2000	0.7471	– 157.8610	6.2000	0.0231	– 222.7626
1.4000	0.5752	– 163.6822	6.4000	0.0213	– 223.9048
1.6000	0.4536	– 168.8713	6.6000	0.0197	– 224.9981
1.8000	0.3649	– 173.5292	6.8000	0.0182	– 226.0453
2.0000	0.2983	– 177.7434	7.0000	0.0169	– 227.0492
2.2000	0.2473	– 181.5850	7.2000	0.0156	– 228.0120
2.4000	0.2075	– 185.1110	7.4000	0.0145	– 228.9362
2.6000	0.1759	– 188.3662	7.6000	0.0135	– 229.8238
2.8000	0.1505	– 191.3866	7.8000	0.0126	– 230.6768
3.0000	0.1298	– 194.2011	8.0000	0.0118	– 231.4969
3.2000	0.1127	– 196.8331	8.2000	0.0110	– 232.2861
3.4000	0.0985	– 199.3019	8.4000	0.0103	– 233.0458
3.6000	0.0866	– 201.6235	8.6000	0.0097	– 233.7776

(Contd.)...

3.8000	0.0765	– 203.8117	8.8000	0.0091	– 234.4828
4.0000	0.0680	– 205.8780	9.0000	0.0086	– 235.1629
4.2000	0.0606	– 207.8325	9.2000	0.0080	– 235.8190
4.4000	0.0543	– 209.6840	9.4000	0.0076	– 236.4523
4.6000	0.0488	– 211.4402	9.6000	0.0072	– 237.0639
4.8000	0.0440	– 213.1080	9.8000	0.0068	– 237.6549
5.0000	0.0398	– 214.6937	10.0000	0.0064	– 238.2262

Table 8.7 Performance parameters for Example 8.5

Parameter	*Un-compensated system*	*Lead-compensated system*	*Lag-compensated system*
PM	– 10.5°	28.7°	50°
GM	– 4.4 dB	11.5 dB	13.9dB
ω_p	2.24 rad/sec	2.12 rad/sec	3.94 rad/sec
ω_g	2.86 rad/sec	1 rad/sec	1.4 rad/sec
BW	3.34 rad/sec	1.24 rad/sec	1.83 rad/sec
Peak response, M_p	unstable	1.48	1.21
Peak time	–	3.09 sec	1.98 sec
% Overshoot	–	48.4	20.7
Rise time, t_r	–	1.13 sec	0.823 sec
Settling Time (2%), t_s	–	15.5 sec	11.9 sec

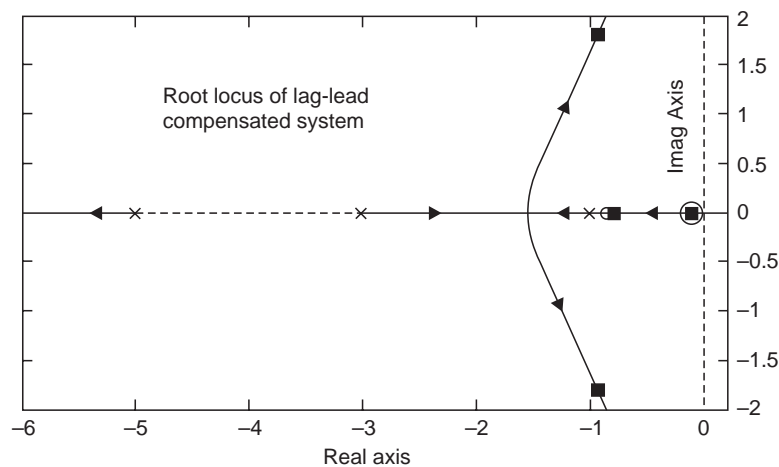



Fig. 8.15 Root locus for $D_{c1}(s)D_{c2}(s)G_p(s) = \dfrac{25.3135(s+0.1023)(s+0.8566)}{s(s+0.0147)(s+1)(s+3.022)(s+5)}$

The lag-lead compensator can be implemented by the RC network of Fig. 8.14 with $C_1 = C_2 = 1$ micro farad, $R_1 = 1.1674$ M ohm, $R_2 = 9.7752$ M ohm, K = A = 0.5069.

8.3 DESIGN OF COMPENSATOR BY ROOT LOCUS TECHNIQUE

When the specification is given in time-domain, the root locus approach is normally used for designing the compensator. The location of the closed loop poles determines the time-domain response and a pair of dominant complex conjugate pole pairs dominate the transient response.

8.3.1 Design of Phase-lead Compensator using Root Locus Procedure

Let the dominant closed loop poles, chosen to meet the time-domain specifications of damping ratio, rise time and / or settling time, be located at s_1 as shown in Fig. 8.16. We join s_1 from the origin O in the s-plane and draw s_1H parallel to the real axis in the s-plane. The line s_1Q is drawn bisecting the angle Os_1H. Let θ be the angle to be contributed by the phase-lead network so that s_1 lies on the root locus of the compensated system. The lines s_1N and s_1R are drawn so that angle Qs_1N = angle Qs_1R = θ/2. The phase-lead compensator pole is then given by the length ON and phase-lead zero is given by OR. The total angle contributed at the desired closed loop pole s_1 by the zero and pole of the compensator is given by

$$\angle ORs_1 - \angle ONs_1 = \angle Ns_1R = \angle\theta \tag{8.23}$$

The phase-lead compensator can, therefore, be written as $D_c(s) = \dfrac{K_c(s + a_1)}{s + b_1}$ (8.24)

where a_1 = RO and b_1 = NO in Fig. 8.16 and K_c may be set to $\alpha = b_1/a_1$ or merged with plant gain K and found by satisfying the amplitude condition of root locus of $D(s_1)G_p(s_1)$ at the desired closed loop pole $s_1 = -\sigma_1 \pm j\omega_1$. With reference to Fig. 8.16 we note, ϕ is the angle the line Os_1 makes with negative real axis and θ is the phase-lead angle of the compensator. Now from elementary geometry, we can show that

$$RM = \omega_1 \tan\left(\frac{\phi - \theta}{2}\right), \quad \text{and} \quad NM = \omega_1 \tan\left(\frac{\phi + \theta}{2}\right)$$

so that
$$a_1 = RM + MO = \sigma_1 + \omega_1 \tan\left(\frac{\phi - \theta}{2}\right) \tag{8.25}$$

and
$$b_1 = NM + MO = \sigma_1 + \omega_1 \tan\left(\frac{\phi + \theta}{2}\right) \tag{8.26}$$

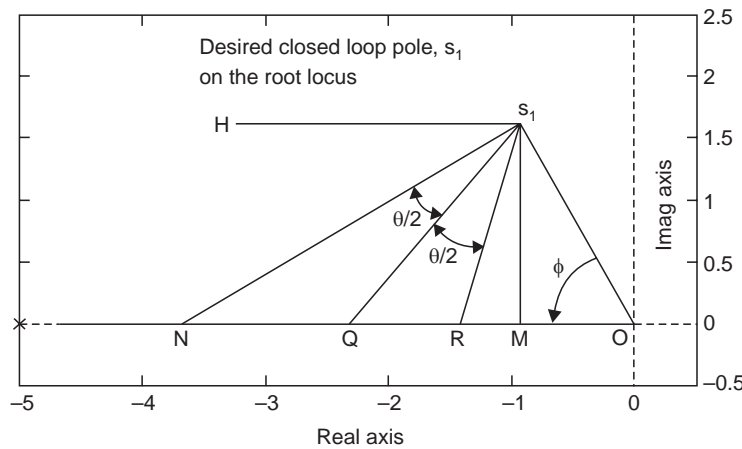



Fig. 8.16 Design of phase-lead compensator by root locus technique

Example 8.5 Let us consider the plant in Example 8.4 again and suppose that from the requirements of time-domain response we choose the closed loop system pole at $s_1 = -1 + j2$.

Here, $G(s) = \dfrac{K}{s(s+1)(s+5)}$.

We find $\angle G(s_1) = -233.1301°$, so the required $\theta = 53.1301°$. Also for s_1 as given above $\phi = 63.4349°$.

Therefore, using relation (8.25) and (8.26) the compensator zero a_1 and pole b_1 are found as

$$a_1 = -1.1803 \quad \text{and} \quad b_1 = -4.2361$$

The total forward path gain K_1 is found from the amplitude condition of root locus as

$$K_1 = K_c K = \frac{|s_1||s_1+1||s_1+5||s_1+4.2361|}{|s_1+1.1803|} = 37.8885$$

If the compensator gain K_c is set at $\alpha = b_1/a_1 = 3.5890$, the plant gain K may be fixed at $K = 37.8885/3.5890 = 10.5568$.

So, $$D_c(s) = \frac{3.5890(s+1.1803)}{s+4.2361}$$

and $$D(s)G(s) = \frac{37.8885(s+1.1803)}{s(s+1)(s+5)(s+4.2361)} \tag{8.27}$$

The transfer function of the closed loop system with unity feedback is given by

$$M(s) = \frac{37.8885(s+1.1803)}{(s+6.949)(s+1.287)(s+1+j2)(s+1-j2)}$$

We find from the transfer function M(s) that the phase-lead compensator could place the closed loop poles exactly at the designed value. It is to be remembered, however, that an arbitrary placement of the closed loop poles may not be realizable all the time. The root locus diagram of D(s)G(s) is shown in Fig. 8.17. The important parameters for the closed loop system with unity feedback when subjected to a unity step input is shown in the second column of Table 8.8.

If it so happens that the desired pole s_1 is such that the required θ becomes negative, the compensator designed based on the relations (8.25) and (8.26) will be a lag network. For example, if the desired pole in Example 8.5 were placed at $s_1 = -0.3 + j1$, the angle θ would be $-6.2813°$, which would have yielded a compensator $D_c(s)$ given by

$$D_c(s) = \frac{0.8492(s+1.1329)}{s+0.9621}$$

So, we find that the design relations (8.25) and (8.26) are equally applicable to lead as well as lag networks. However, since the lag network is intended to improve the steady state response, rather than transient response, we shall follow the procedure given below for designing lag compensator by root locus technique.

In Example 8.5, the phase-lead compensator has been designed to meet the transient response of the closed loop system. However, the velocity error constant K_v in the above example is found to be only 2.1108 (vide Table 8.8). So, the steady state error for ramp input will be large. We are interested to increase the value of velocity error constant by a factor of 8 to 10. This can be done by the phase-lag compensator.

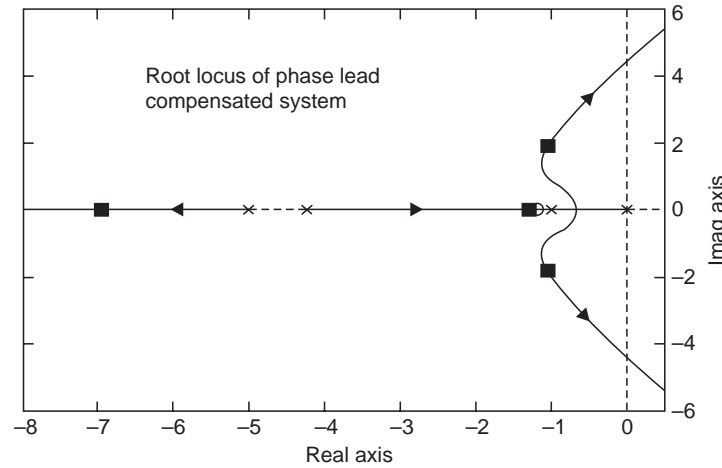



Fig. 8.17 Root locus of phase-lead compensated system in Example 8.5

8.3.2 Design of Phase-lag Compensator using Root Locus Procedure

Let us consider the transfer function of the phase-lag compensator in Equation (8.10) after renaming it as $D_{c1}(s)$

$$D_{c1}(s) = \frac{1}{\alpha}\left(\frac{s + a_1}{s + b_1}\right) \tag{8.28}$$

If this lag compensator is placed in series with the forward path of an uncompensated system with type 1 transfer function the velocity error constant, K_v, can be written as

$$K_v = \lim_{s\to 0} sG(s)D_{c1}(s) = \left[\lim_{s\to 0} sG(s)\right]\left[\lim_{s\to 0} D_{c1}(s)\right]$$

$$= [K_{vu}]\left[\lim_{s\to 0} \frac{1}{\alpha}\frac{s + a_1}{s + b_1}\right] = \left(\frac{K_{vu}}{\alpha}\right)\left(\frac{a_1}{b_1}\right) \tag{8.29}$$

where K_{vu} is the velocity error constant of the plant without the lag compensator. So, we see from (8.29) that K_{vu} can be increased by a factor of α and steady state error due to ramp input to the plant may be decreased by the same factor. It is to be remembered that the attenuation in the lag compensator is simply transferred to the plant gain, the overall loop gain remains unchanged and consequently, the closed loop poles and hence the transient response of the system should remain unaffected by phase-lag compensator design. This is illustrated by the following example.

Example 8.6 We found that the lead compensated system in Example 8.5 has K_v = 2.1108 and we are interested to increase it by a factor of 8 without disturbing the transient response. So, we take α = 8 and choose the zero and pole of the lag compensator very close to the origin. A commonly used figure. for the zero is a_1 = 0.1, hence b_1 = 0.1/8 = 0.0125. Therefore, transfer function of the lag compensator is given by

$$D_{c1}(s) = \frac{0.125(s + 0.1)}{s + 0.0125}$$

Since, both the zero and pole is very close to the origin, the phase angle contributed by them at the desired pole of the closed loop system will be negligibly small (< 2° in the present case) such that the root locus will almost remain unchanged. The step response

parameters of the closed loop system is shown in Table 8.8 with unity feedback and loop gain $G(s)D(s)D_{c1}(s)$ given by

$$G(s)D(s)Dc_1(s) = \frac{303.108(s+1.1803)}{s(s+1)(s+5)(s+4.2361)} \frac{0.125(s+0.1)}{s+0.0125}$$

The closed loop transfer function is given by

$$M(s) = \frac{37.8885(s+1.1803)(s+0.1)}{(s+6.949)(s+1.3)(s+0.1043)(s+0.9535+j1.9611)(s+0.9535-j1.9611)}$$

From the Table 8.8, we find that the transient response remains almost unchanged, but the velocity error constant K_v of the plant could be increased by factor of 8.

Table 8.8

Parameter	*Lead-compensated system*	*Lag-lead compensated system*
Peak response, M_p	1.23	1.25
Peak time	1.7 sec	1.72 sec
% Overshoot	23.3	25.5
Rise time, t_r	0.675 sec	0.675 sec
Settling Time, t_s	3.87 sec	7.9 sec
K_v	2.1108	16.891

8.4 PID CONTROLLER

The transfer function of 3-term PID controller is given by:

$$D(s) = K_p + K_d s + \frac{K_i}{s} \tag{8.30}$$

This can be expressed in the form

$$D(s) = \frac{K_d(s+a_1)(s+a_2)}{s} \tag{8.31}$$

which indicates that PID controller is similar to lag lead compensator, with one absent pole. It is not possible to realize the PID controller by passive RC network because of the pure integration term and one more zero than the pole in its transfer function . However, it may be realized by active electronic components, Hydraulic and Pneumatic components.

Since, the PID controller is similar to lag lead compensator, its design may be undertaken by analytical methods when the mathematical model of the plant is available. However, mathematical models of some plants may not be known accurately and for some, it is difficult to get. The model of an electrical oven, for instance, can be prepared in principle by measuring the electrical parameters of the heating coil, thermal resistance and thermal capacity. The physical mechanism responsible for loss of heat like radiation, conduction, convection are also understood, but difficult to measure accurately for a given unit. The transfer characteristic between the electrical input of an oven and the steady state temperature, on the other hand, can be measured experimentally, rather easily. When the analytical model is not available, analytical design of PID controller is not possible. Ziegler and Nichols have proposed a procedure for selecting the parameters of the PID controller when the plant model is measured experimentally.

8.4.1 Ziegler-Nichols Rules for Tuning PID Controllers

Ziegler and Nichols [29] proposed rules for determining values of the proportional gain K_p, integral gain K_i, and derivative gain K_d based on the transient response characteristics of a given plant. Such determination of the parameters of PID controllers referred to as tuning of PID controllers can be made by engineers on site by experiments on the plant.

There are two methods known as tuning rules of Ziegler-Nichols. In both the methods, Ziegler and Nichols aimed at obtaining 25% maximum overshoot in step response.

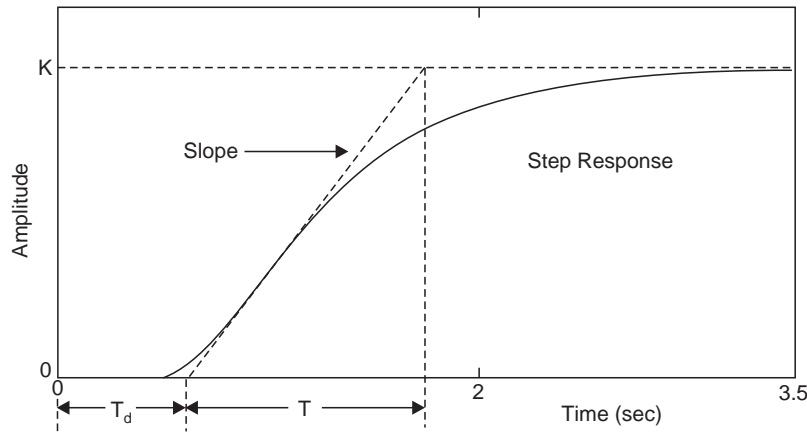



Fig. 8.18 Unit-step response of a process with time delay T_d

8.4.2 First Method

In the first method, the response of the open loop plant subjected to a unit-step input is obtained experimentally, as shown in Fig. 8.18. If the plant does neither contain pure integrators nor dominant complex-conjugate poles, then the unit-step response curve may look like an S-shaped curve as shown in Fig. 8.18. This method is not applicable, if the response does not exhibit an S-shaped curve. The step-response curve may also be generated from a simulation of the dynamics of the plant to apply the first method.

The S-shaped curve may be characterized by two constants, delay time T_d and plant time constant T. The delay time and time constant are determined by drawing a tangent line at the inflection point of the S-shaped curve (vide Fig. 8.18) and determining the intersections of the tangent line with the time axis and the line corresponding to the steady state value $y(t) = K$. The transfer function $Y(s)/R(s)$ may then be approximated by a first-order system with a transportation lag T_d .

$$\frac{Y(s)}{U(s)} = \frac{Ke^{-T_d s}}{Ts + 1} \tag{8.32}$$

Ziegler and Nichols suggested the values of the parameters K_p, K_i and K_d according to the formula shown in Table 8.9. So the PID controller tuned by the first method of Ziegler-Nichols rules is given by

$$D_c(s) = K_p + \frac{K_i}{s} + K_d s = K_p \left(1 + \frac{k_i}{s} + k_d s\right)$$

Table 8.9 Ziegler-Nichols Tuning Rule Based on Step Response of Open Loop Plant

Type of controller	K_p	k_i	k_d
P	$\frac{T}{T_d}$	0	0
PI	$\frac{0.9T}{T_d}$	$\frac{0.3}{T_d}$	0
PID	$\frac{1.2T}{T_d}$	$\frac{1}{2T_d}$	$0.5T_d$

$$= \frac{1.2T}{T_d}\left(1 + \frac{1}{2T_d s} + 0.5T_d s\right) = 0.6T\,\frac{\left(s + \frac{1}{T_d}\right)^2}{s} \qquad (8.33)$$

Thus, the PID controller has a pole at the origin and double zeros at $s = -1/T_d$.

8.4.3 Second Method

In the second method, we first set $k_i = 0$ and $k_d = 0$. Using only the proportional control, the forward path gain K_p is increased from 0 to a critical value K_{pc} when the output first exhibits sustained oscillations. In case the output does not exhibit sustained oscillations for any forward path gain K_p, then this method is not applicable. Thus, the critical gain K_{pc} and the corresponding period T_c are experimentally determined (see Fig. 8.19). Ziegler and Nichols suggested the values of the parameters K_p, k_i and k_d according to the relations shown in Table 8.10.

Table 8.10 Ziegler-Nichols Tuning Rule Based on Critical Gain K_{pc} and Critical Period T_c (Second Method)

Type of controller	K_p	k_i	k_d
P	$0.5\ K_{pc}$	0	0
PI	$0.45\ K_{pc}$	$\frac{1.2}{T_c}$	0
PID	$0.6\ K_{pc}$	$\frac{2}{T_c}$	$0.125\ T_c$

From Table 8.10, the PID controller, tuned by the second method of Ziegler-Nichols rules, is, therefore, found to be of the form

$$D_c(s) = K_p\left(1 + \frac{k_i}{s} + k_d s\right) = 0.6\ K_{pc}\left(1 + \frac{2}{T_c s} + 0.125T_c s\right) = 0.075\ K_{pc}\ T_c\,\frac{\left(s + \frac{4}{T_c}\right)^2}{s} \qquad (8.34)$$

Thus, the PID controller has a pole at the origin and double zeros at $s = -4/T_c$.

The tuning rules presented by Ziegler and Nichols have been widely used over the last few decades in process control systems where the plant dynamics are not known accurately.

Even though, the tuning rules may also be used for the design of PID controllers in plants with known dynamics, availability of alternative analytical and graphical tools are preferred to Ziegler-Nichols tuning rules.

If the transfer function of the plant is known, a unit-step response for the critical gain K_{pc} and critical period T_c may be calculated. Then, using those calculated values, it is possible to determine the parameters K_p, k_i and k_d from Table 8.9 or 8.10. However, the real usefulness of Ziegler-Nichols tuning rules becomes apparent when the plant dynamics are not known so that no analytical or graphical approaches to the design of controllers are available. However, if the closed-loop does not exhibit sustained oscillations, the critical gain K_{pc} as well as critical period T_c does not exist. In such cases the second method is not applicable.

The plant with a PID controller tuned by Ziegler-Nichols rules, when it is applicable, will exhibit approximately 10% ~ 60% maximum overshoot in step response. In a given situation, if the maximum overshoot is not acceptable, fine tuning of the controller parameters are inevitable for satisfactory transient responses of the closed-loop system. In this respect, Ziegler-Nichols tuning rules provides, an educated guess for the parameter values and gives a starting point for fine tuning.

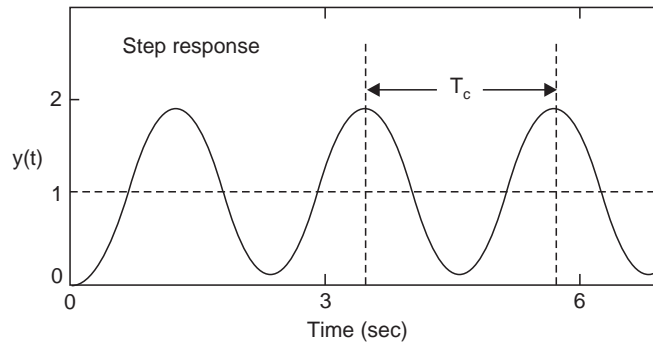



Fig. 8.19 Critical period of oscillation for PID controller tuning

Example 8.7 Consider the control system shown in Fig. 8.9(*a*) with plant transfer function given by:

$$G_p(s) = \frac{25}{s(s+2)(s+5)}$$

and $D_c(s)$ is the PID controller with transfer function of the form

$$D_c(s) = k_p \left(1 + \frac{k_i}{s} + k_d s\right)$$

We shall illustrate the Ziegler-Nichols tuning rule for the determination of the parameters K_p, k_i and k_d, even though any of the analytical methods developed so far, may be employed for the design of the PID controller. Since the plant has an integrator, we use the second method of Ziegler-Nichols tuning rules. With $k_i = 0$ and $k_d = 0$, the closed-loop transfer function is obtained as:

$$\frac{Y(s)}{R(s)} = \frac{K_p}{s(s+1)(s+5)+K_p}, \text{ where } K_p = 25k_p \tag{8.35}$$

We shall prepare the Routh-Hurwitz Table for computing the critical forward path gain, K_{pc} that makes the system marginally stable and exhibits sustained oscillation with period T_c.

The characteristic equation for the closed-loop system is given by

$$s^3 + 7s^2 + 10s + K = 0$$

and the Routh-Hurwitz table for the above characteristic equation is shown below:

s^3	1	10
s^2	7	K
s^1	$\frac{70-K}{7}$	0
s^0	K	0

Examining the coefficients in the first column of the Routh table, we find the critical gain for sustained oscillation to be $K_{pc} = 70$. The auxiliary equation formed with entries corresponding to the row of s^2 becomes

$$7s^2 + K = 7s^2 + 70 = 0$$

This yields the frequency of oscillation, $s = \pm j\omega = \pm j\sqrt{10} = \pm j3.1623$

Hence, the period of sustained oscillation is

$$T_c = \frac{2\pi}{\omega} = \frac{2\pi}{\sqrt{10}} = 1.9869 \text{ sec}$$

The parameters of the PID controller are computed using the relations in Table 8.10 as:

$$K_p = 0.6\, K_{pc} = 42;$$

so that k_p is found using (8.35) as $k_p = 42/25 = 1.68$

and $$k_i = 2/T_c = 1.007,\ k_d = 0.125T_c = 0.2484$$

The transfer function of the PID controller is, therefore, given by

$$D_{c1}(s) = k_p\left(1 + \frac{k_i}{s} + k_d s\right) = 1.68\left(1 + \frac{1.007}{s} + 0.2484s\right) = \frac{0.4173(s + 2.0134)^2}{s}$$

The PID controller has a pole at the origin and double zero at $s = -2.0134$.

The step response of the closed loop system with unity feedback and the PID controller designed above by Ziegler-Nichols tuning method gives rise to 60% overshoot and needs fine tuning. Since, the over shoot is to be decreased , k_d has to be increased. As a trial solution, let the double zero be relocated at – 1.0 and setting $k_p = 1$ gives rise to the following PID controller.

$$D_{c2}(s) = \frac{(s+1)^2}{s} \tag{8.36}$$

The over shoot in this case is found to be 22.4%. The summary of step response is presented in Table 8.11.

Table 8.11 Step Response of PID Controller for System of Example 8.7

Parameter	$D_{c1}(s)$	$D_{c2}(s)$
Peak response, M_p	1.60	1.22
Peak time	1.18 sec	0.765 sec
% Overshoot	60	22.4
Rise time, t_r	0.423 sec	0.321 sec
Settling Time, t_s	5.82 sec	3.63 sec

8.5 DESIGN OF COMPENSATORS FOR DISCRETE SYSTEMS

Consider the series compensator for discrete system as shown in Fig. 8.20. The closed loop transfer function for the system in Fig. 8.20 is given by:

$$\frac{Y(z)}{R(z)} = \frac{D(z)G_p(z)}{1 + D(z)G(z)} \tag{8.37}$$

and hence the characteristic equation is given by:

$$1 + D(z)G(z) = 0 \tag{8.38}$$

where G(z) is the z transform of $G_p(s)H(s)$

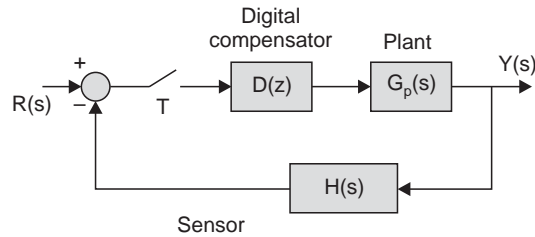



Fig. 8.20 Digital control system

Let us consider transfer function of the discrete compensator in the form:

$$D(z) = K_d \frac{(z - z_o)}{z - z_p} \tag{8.39}$$

In Appendix C it has been shown that the Tustin transformation can be used to map z-domain to w-domain which is very similar to s-plane. So, the design relations (8.16) and (8.17) for analog lag-lead compensators can also be used for lag and lead compensators for discrete systems where the frequency response G($j\omega$) is to be substituted by the frequency response of the discrete system in the w-domain. This can be easily verified by considering the Tustin transformation

$$w = \frac{2}{T}\frac{z - 1}{z + 1} \tag{8.40}$$

Therefore, the compensator in (8.39) may be written in the w-domain as

$$D_c(w) = \frac{K_c(w + a_1)}{(w + b_1)} \tag{8.41}$$

where $K_d = \dfrac{b_1}{a_1}\dfrac{a_1 + \dfrac{2}{T}}{b_1 + \dfrac{2}{T}}$, $z_o = \dfrac{\dfrac{2}{T} - a_1}{\dfrac{2}{T} + a_1}$, and $z_p = \dfrac{\dfrac{2}{T} - b_1}{\dfrac{2}{T} + b_1}$, $K_c = \dfrac{b_1}{a_1}$ (8.42)

The frequency response in the w-domain for the digital compensator is written as

$$D_c(j\omega_w) = \frac{\alpha_0 + j\alpha_1\omega_w}{1 + j\beta_1\omega_w} \quad (8.43)$$

where $a_1 = \dfrac{\alpha_0}{\alpha_1}$, $b_1 = \dfrac{1}{\beta_1}$, $K_c = \dfrac{\alpha_1}{\beta_1}$ and ω_w is the frequency in the w-domain.

We are now able to directly utilize all the design relations developed in connection with the equivalent analog compensators for the closed loop system shown in Fig. 8.9(b).

The compensator parameters may be written in terms of the frequency response of $G(j\omega_w)$ at the gain crossover frequency ω_{w1}, as shown below.

$$\beta_1 = \frac{\cos\theta - \alpha_0 |G(j\omega_{w1})|}{\omega_{w1} \sin\theta} \quad (8.44)$$

$$\alpha_1 = \frac{1 - \alpha_0 |G(j\omega_{w1})| \cos\theta}{\omega_{w1} |G(j\omega_{w_1})| \sin\theta} \quad (8.45)$$

The plant transfer function $G_p(s)$ in Fig. 8.9(a) is preceded by a zero order hold circuit and the characteristic equation of the closed loop system is given by

$$1 + D_c(z)\, G(z) = 0 \quad (8.46)$$

where $G(z)$ is the z transform of $\left(\dfrac{1 - e^{-Ts}}{s}\right) G_p(s)$

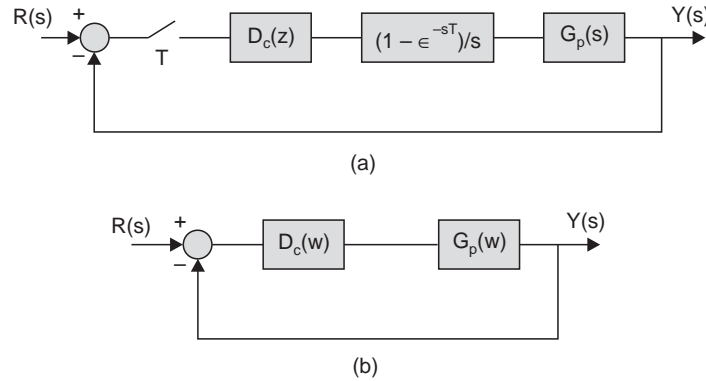



Fig. 8.21 Discrete compensator for a continuous system (a) Series compensator followed by zero order hold and analog plant (b) Equivalent analog model

From the equivalent analog system in w-domain as shown in Fig. 8.9(b), the characteristic equation of the closed loop system is given by

$$1 + D_c(w)G(w) = 0$$

and from the frequency response with $w = j\omega_w$, we can write $D(j\omega_w) = -1/G(j\omega_w)$ and proceed to complete the design of the analog compensator elaborated in sec 8.2.3. However, the digital compensator is to be implemented on a microcomputer, unlike the analog one which was realized by passive RC network and an operational amplifier. In order to do this the result, obtained after completing the design in analog domain, is to be transformed back to get the digital compensator D(z) in equation (8.39) by using following relations

$$K_d = \frac{1 + \frac{2}{T}\frac{\alpha_1}{\alpha_0}}{1 + \frac{2}{T}\beta_1}, \; z_o = \frac{\frac{2}{T} - \frac{\alpha_1}{\alpha_0}}{\frac{2}{T} + \frac{\alpha_1}{\alpha_0}}, \text{ and } z_p = \frac{\frac{2}{T} - \frac{1}{\beta_1}}{\frac{2}{T} + \frac{1}{\beta_1}} \quad (8.47)$$

For ready reference the design steps of Section 8.2.4 and 8.2.5 for analog lag and lead compensators respectively, using frequency response of $G(j\omega_w)$, is reproduced below. The frequency response of the plant $G(j\omega_w)$ may be found by Bode plot.

8.5.1 Design Steps for Lag Compensator

Let θ be the phase angle of $D_c(w)$ at the gain crossover point for the Lag Network, which is negative.

Step 1: Find ω_{w1} from frequency response of $G(j\omega_{w1})$, such that

$$\angle G(j\omega_{w1}) \geq -180° + \phi_m - \theta, \; -5° \geq \theta > -10°.$$

For a stable compensator β_1 should be positive and, since θ and sin θ are both negative, so that we can write step 2 from relation (8.44)

Step 2: $\alpha_0 |\, G(j\omega_{w1}) \,| > \cos\theta$.

8.5.2 Design Steps for Lead Compensator

For Lead Network θ is positive

Step 1: Find ω_{w1}, from frequency response of $G(j\omega_w)$, such that

$$\angle G(j\omega_{w1}) \leq -180° + \phi_m - \theta, \text{ for } 70° \geq \theta > \phi_m + 5° - \phi_{mu},$$

where, ϕ_{mu} is the phase margin of the uncompensated system.

For a stable compensator β_1 should be positive. Hence for a positive θ and sin θ we must have.

Step 2: $\cos\theta > \alpha_0 \,|\, G(j\omega_{w1}) \,|$

Example 8.8 Phase-lead Network.

With reference to the system in Fig. 8.9(a) with $G_p(s) = 2/\{s(s+1)(s+2)\}$, find the discrete compensator to achieve a phase margin of 45° when the sampling period is chosen as T = 0.02 sec.

Since the design steps are the same as in the analog case, the details are omitted. The MATLAB Mfile in Script_Example8.8 has been used to get the compensator coefficients in w and z domain as shown below.

$$D_{c1}(w) = (1 + 2.5640w)/(1 + 0.2747w)$$

and $$D_{c1}(z) = 9.04(z - 0.9922)/(z - 0.9298)$$

Once the filter coefficients are found, the Bode diagram (or Nyquist diagram) must be obtained to check the stability of the closed-loop system. With H(z) = 1, for the above system, the parameters of interest we found as: GM = 10.1 dB, PM = 45°, phase cross over frequency ω_{wg} = 3.1636 and gain cross over frequency ω_{wp} = 1.5999. So the closed loop system is stable and the designed compensator meets the PM specifications.

Example 8.9 Design of Lag Compensator.

Consider again the system in Example 8.8, where the phase margin of ϕ_m = 45° is to be achieved using a unity-gain phase-lag compensator. The MATLAB Mfile in Script_example8.9 is used to get the compensator coefficients shown below:

$$D_{c2}(w) = (1 + 15.97w)/(1 + 36.71w)$$

and $$D_{c2}(z) = \{0.4352\,(z - 0.9989)\}/(z - 0.9995)$$

The frequency domain parameters of interest are:

$$GM = 16.05\ dB,\ PM = 51.7°,\ \omega_{wg} = 1.3565,\ \omega_{wp} = 0.4000$$

So the designed systems is stable and it meets the design specifications.

MATLAB SCRIPTS

```
%Script_Example8.1
%Design of Lead Compensator
clear all; close all hidden
gl=zpk([ ], [0 – 5 – 2],25); w=0.1:0.1:5;
L=length(w);[MAG PHASE]=bode(g1,w);
mag=reshape(MAG,L,1);            %result in Table 8.1
phase=reshape(PHASE,L,1);        %result in Table 8.1
[Gm phimu wg wp]=margin(g1),     %Phase margin of the uncompensated plant
phim=45;                         %theta>21 deg = phim + 5 – phimu
     %find w1 for which phase of G(jw1)<–159 deg = – 180 + θ
w1=2.2                           %select w1 from Table 8.1
g=0.6997; PhaseG= – 161.4758     %Mag and phaseG at w1 = 2.2 rad/sec found
                                 from Table 8.1
theta=–180+phim–phaseG;          % required theta=26.4758
a0=1; n1=g*cos(theta*pi/180); d1=w1*sin(theta*pi/180);
alpha1=(1–a0*n1)/(d1*g)          %ref equ (8.17)
n2=cos(theta*pi/180)-a0*g;
beta1=n2/d1                      % ref equ (8.16)
alpha=alpha1/beta1)
K=alpha1*a0/beta1                %ref to equ (8.3) with K = Kc
a1=1/alpha1, b1=1/beta1
C=power(10, – 6)                 %RC realisation of lead network
R1=alpha1*beta1/C                %Ref to Equ (8.6)
R2=R1/(alpha-1)
gc=zpk([–a1],[–b1]),alpha);
g2=g1*gc;
bode(g2)
```

```
Script_Table8.1
%To prepare table 8_1 in chapter 8 for Example 8.2
clear all;close all hidden
g1=zpk([ ],[0 – 2 – 5],25)
%[Gm,Pm,wg,wp]=margin[g1]
w=0.1:0.1:5;
```

```
L=length(w)
[MAG PHASE] = bode(g1,w);
mag=reshape(MAG, L,1);
phase=reshape(PHASE,L,1);
```

```
%Script_Example8.2
%Design of Lag Compensator
clear all;close all hidden
g1=zpk([ ],[0 –5 –2],25); w=0.1:0.1:5;
L=length(w);(MAG PHASE)=bode(g1, w);
mag=reshape(MAG,L,1);           % result in Table 8.1
phase=reshape(PHASE,L,1);       % result in Table 8.1
phim=45;theta=–5;
            % find w1 for which phase of G(jw1) >
            –180+phim–(– 5)=–130 deg
w1=1.0      % w1 found from Table 8.1
g=2.1926    % Mag at w1 = 1.0 rad/sec found from Table 8.1
     %|G(jw1)| > cos5=0.9962 of step 2 for Lag compensator design
     % is satisfied
a0=1;
n1=g*cos(theta*pi/180);d1=w1*sin(theta*pi/180);
alpha1=(1-n1*a0)/(d1*g)         % ref equ (8.17)
n2=cos(theta*pi/180)-a0*g;
beta1=n2/d1                     % ref equ (8.16)
alpha=beta1/alpha1
K=a0/alpha, a1=1/alpha1, b1=1/beta1
C=power(10, –6), R2=alpha1/(C)  % ref equ (8.10)
R1=R2* (alpha-1),
gc=zpk([–a1],[–b1], K), g2=g1*gc;bode(g2)
```

```
%Script_Example 8_5
%Design of lead compensator using root locus
s=–1+2i; num1=[1];den1=[s^3+6*s^2+5*s];
g=num1/den1;                                    %Plant tf
u=real(g); v=imag(g);phaseG=atan(v/u);Gdeg=–180+180*phaseG/pi
theta=phaseG; thetadeg=abs(Gdeg)–180
u1=(real(s)); v1=imag(s); phi=atan(v1/u1); phideg=abs(180*phi/pi)
x1=0.5*(theta+phi); x2=0.5*(phi-theta);
a1=abs(real(s))+abs(imag(s)*tan(x2))            %Ref Equ (8.25)
b1=abs(real(s))+abs(imag(s)*tan(x1))            %Ref Equ (8.26)
d=(s+a1)/(s+b1);
g1=g*d;
K=1/norm(g1)
g2=zpk([–a1],[–b1], K);
g3=zpk([],[0 –1 –5],1);
g4=g2*g3
```

```
g5=feedback(g4,1)              %Unity fb sys
%For non-intuitive visualisation use % phaseG=atan2(v,u)
%phi=atan2(v1,u1);and x1d=0.5*(thetadeg+phideg);
%x2d=0.5*(phideg-thetadeg); %x1=x1d*pi/180; x2=x2d*pi/180
%Problem:show that closed loop pole located at s=–0.3+1i
will give rise to lag network
```

```
%Script_example8.8
% Design of lead compensator for discrete system
T=0.02;a0=1;g1-zpk([],[0 –1 –2], 2); g2=c2d(g1,T);
[Gmu,Pmu,Wgu,Wpu] = margin(g2);
w=0.1:0.1:5;                          %arrange in tabular form
L=length(w);[MAG  PHASE]=bode(g2,w);
mag=reshape(MAG,L,1);                 % arrange in tabular form
phase=reshape(PHASE,L,1);             % arrange in tabular form
phim=45°;                             %theta > 17.8204=phim+5–Pnu
                                      %find w1 such that angleG(jw1)
                                            <=–180+theta
                                      % w1 > 0.9 found from table
w1=1.6; angleG=–187.5712; G=0.2587;
theta=(–180+phim–angleG)*pi/180
alpha1=(1-a0*G*cos(theta))/(w1*G*sin(theta));
beta*pi/1801=(cos(theta)-(a0*G))/(w1*sin(theta));
num1=[alpha1 a0];den1=[beta1 1];d1=tf(num1,den1),
d2=c2d(d1, T,'tustin') %discretized compensator with Tustin transformation
%d2=c2d(d1, T,'prewarp',1/beta1) Prewarping also can be used
g4=series(g2,d2);
[Gm,Pm,Wg,Wp] = margin(g4)
bode(g2,'k–',g4,'r',w)
```

```
%Script_Example8.9
% Design of lag compensator for discrete system
clear all; close all hidden;
T=0.02;a0=1; g1=zpk([],[0 –1 –2],2); g2=c2d(g1,T);
[Gmu,Pmu,Wgu,Wcu] = margin(g2);
w=0.1:0.1:5;                          %arrange in tabular form
L=length(w);[MAG PHASE]=bode(g2,w);
mag=reshape(MAG, L, 1);               % arrange in tabular form
phase=reshape(PHASE,L,1);             % arrange in tabular form
theta=–5; phim=45
%find w1 such that angleG(jw1) >=–180+phim–theta= –130 deg
% w1 < 0.5 found from table
w1=0.4; angleG=–123.3405; G=2.2761;
theta=(theta*pi)/180;
alpha1=(1–a0*G*cos(theta))/(w1*G*sin(theta));
beta1=(cos(theta)–(a0*G))/(w1*sin(theta));
```

```
num1=[alpha1 a0];den1=[beta1 1];d1=tf(num1,den1)
d2=c2d(d1,T,'tustin') % discretized compensator with Tustin transformation
%d2=c2d(d1,T,'prewarp', 1/beta1) Prewarping also can be used
g4=series(g2,d2);
[Gm,Pm,Wg,Wp] = margin(g4)
bode(g2, 'k–',g4,'r', w)
```

REVIEW EXERCISE

RE8.1 With reference to the system in Fig. 8.9(*a*) let G(*s*) be given by:

$$G(s) = \frac{4K}{(s+2)(s+4)}$$

Design a lag compensator to meet the following specification:

(*a*) Position error constant $K_p \geq 100$ and (*b*) phase margin ≥ 45°.

Ans: $G_c(s) = \dfrac{0.0568(s+0.5568)}{s+0.0316}$, **Hints:** K ≥ 200 to satisfy the specification on K_p. Set K = 200 and tabulate frequency response. At $\omega_1 = 6$, $|G(j\omega_1)| = 17.5412$ satisfies the conditions of design steps of lag network. The frequency-domain parameters for the compensated system are GM = ∞, phase margin = 47.1° , $\omega_g = \infty$, $\omega_p = 6.002$. The closed loop system with H(*s*) = 1, when subjected to a step input yields an overshoot of 15.6% and settling time $t_s = 3.25$ sec.

RE8.2 Design a lead compensator for the problem in 8.1.

Ans: $G_c(s) = \dfrac{4.9928(s+20.9607)}{s+104.6527}$

Hints: As before, set K = 200 and tabulate the frequency response. At $\omega_1 = 40$, $|G(j\omega_1)| = 0.4969$ satisfies the conditions of design steps of lead network. The frequency-domain parameters for the compensated system are GM = ∞, phase margin = 50°, $\omega_g = \infty$, $\omega_p = 40$. The closed loop system with H(*s*) = 1, when subjected to a step input yields an over shoot of 24.9% and settling time $t_s = 0.146$ sec.

RE8.3 Design a PID controller for the dc motor position control system with system parameter [see equation A.71 in Appendix A] given by:

Moment of inertia,	$J = 10.0 \times 10^{-2}$	Kg-m/rad/sec^2
Coefficient of viscous friction,	$B = 5.0 \times 10^{-3}$	Kg-m/rad/sec
Motor constant,	KT = 0.5	Kg-m/Amp
Armature resistance,	Ra = 5.0	ohms
Armature inductance,	$La = 2.0 \times 10^{-3}$	henry
Back emf,	Kb = 0.65	volts/rad/sec.

The closed loop system is required to meet the specifications:

overshoot	< 10 %
rise time,	$tr < 1$ sec
settling time	$ts < 3$ sec
steady state error	$e_{ss} = 0$.

The PID controller can be designed using MATLAB/Simulink. A script file is given below for PID configuration. It may be modified for proportional and proportional-derivative configuration

```
J = 10.0e-2
B = 5.0e-2
KT = 0.5
Ra = 5.0
La = 2.0e-2
Kb = 0.65
num = KT ;
den = [(J*La) ((J*Ra) + (La*B)) ((B*Ra) + (Kb*KT)) 0];
g = tf(num,den);
t = 0.0:0.05:12;
Kp = 5;
Kd = 2.0;
Ki = 0.001;
num1 = [Kd Kp Ki];
den1 = [1 0]
h = 1
num2 = conv(num1,num)
den2 = conv(den1,den)
g2 = tf(num2,den2);
gcloop = feedback(g2, h);
step(gcloop, t);
% For proportional control make the following changes in the script:
num1 = Kp;
den1 = 1;
num2 = num1*num;
den2 = den1*den;
% Ans: (a) Kp = 4; overshoot = 39.4%, tr = 0.651 sec, ts = 6.87 sec, ess = 0
% (b) Kp = 4, Kd = 2.0: overshoot = 5.79%, tr = 0.692 sec, ts = 2.5 sec, ess = 0
% (c) Kp = 4, Kd = 2.50, Ki = 0.075: overshoot = 6.33%, tr = 0.689 sec, ts = 2.65 sec, ess = 0
% (d) Kp = 5, Kd = 2.0, Ki = 0.001: overshoot = 9.33%, tr = 0.595 sec, ts = 2.23 sec, ess = 0
```

For this system, the steady state error e_{ss} is zero in all the combinations of K_p, K_d, and K_i given above, including the cases where $K_i = 0$. We note that the solution in (*a*) fails to meet the design specifications, other than that on e_{ss}. Therefore, so far as meeting the present specifications are concerned, the solution in (*b*) with PD controller is good enough. However, in a practical system involving mechanical motion, many un-modeled nonlinearities like stiction and coulomb friction will be present, where the inclusion of integral term is advisable to take care of these un-modeled frictions.

PROBLEMS

P8.1 The step response of the lag compensated system in RE8.1 shows that the response is sluggish (t_s = 3.25 sec) whereas the response of the lead compensated system is quite fast but accompanied with large overshoot. Therefore, design a lag-lead compensator for the plant in RE8.1 to meet the following specification: $K_p \geq 100$, phase margin $\geq 45°$, $t_s \leq 1$ sec and overshoot ≤ 10 %.

P8.2 For the system in Fig. 8.9(*a*) let G(*s*) be given by:

$$G(s) = \frac{K}{s(s+1)}$$

Design a lead compensator to meet the specification: $K_v \geq 100$, $\phi_m \geq 50°$.

P8.3 Design a lag-lead compensator to meet the specification: $K_a \geq 10$, $\phi_m \geq 55°$ for the system 8.9(*a*) if the plant transfer function G(*s*)is given by: $G(s) = \dfrac{10K}{s^2(s+2)}$.

P8.4 The lead network in RE8.2 was found using a cascade compensator $G_c(s)$ in the system configuration shown in Fig. 8.9 (*a*) in order to meet the specification on K_p and ϕ_m. The same lead network would be obtained if we used the feedback compensation configuration of Fig. 8.1(*b*) with H(s) = 1. Determine the closed-loop transfer function M(*s*) = Y(*s*)/R(*s*) of both the cascade and feedback configurations and show how the transfer function of each configuration differs. Find the overshoot and the settling time (2% criteria) to a step response. Can you explain the difference in the response?

Ans: overshoot 2.63%, t_s = 0.153 sec.

P8.5 The plant transfer function G(*s*) of a unity feedback system is given by:

$$G(s) = \frac{K}{s(s+2)}$$

Design a lead compensator in order to place closed loop poles at $s = -3 \pm j4$. Also find the over shoot of the closed loop system with unity step input.

P8.6 Find K_v for the compensated system of problem P8.5 for a ramp input with unity slope. Introduce a lag compensator in cascade so as to increase K_v by a factor of 5 and find its parameters.

P8.7 The block diagram of a turner lathe control system is shown in Fig. P8.7. The servo control system for positioning the cutting tool head is represented by $G_p(s)$ and $G_1(s)$ represents the power amplifier and *n* is the gear ratio. In order to attain an accuracy of one thousandth of an inch in the tool position for a ramp input $K_v \geq 50$ and damping ratio of the dominant closed loop poles is 0.7. Design a cascade lag compensator to be inserted in front of $G_1(s)$.

Given $\quad G_p(s) = \dfrac{1000}{s(s+10)}$ and $G_1(s) = \dfrac{50}{s+10}$

Fig. P8.7 Turner lathe control system

P8.8 The Block diagram of a chemical process whose production rate is a function of catalytic agent is shown in Fig. P8.8. The transfer function $G_p(s)$ of the reactor is given below where the time delay is 40 sec. Design a compensator by frequency-domain method which will ensure a phase margin ≥ 35° and steady state error to a step input less than 0.1. Also find the settling time of the closed loop system.

Given $\quad G_p(s) = \dfrac{e^{-sT}}{(25s+1)^2}$

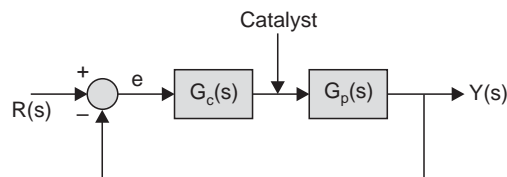



Fig. P8.8 Block diagram of a chemical process

CHAPTER

9

State Feedback Design

9.1 POLE ASSIGNMENT DESIGN AND STATE ESTIMATION

The correlation of the closed loop poles with transient response has been discussed in Chapter 4 and in Chapter 7. The nature of step-response of a second order system for some typical location of the closed loop poles in the s-domain have been shown in Fig. 4.6 and Fig. 4.7 of Chapter 4.

We shall now develop a design procedure for generating control laws generally known as pole assignment or pole placement, when the system dynamics is represented in state variable form. The design procedure is concerned with the assignment of the poles of the closed-loop transfer function to any desired locations in the stable zone of the plane of the roots by using state variable feedback.

In order to introduce the pole assignment technique, we consider the transfer function of armature controlled D.C. motor shown in Fig. 9.1. With a choice of states $x_1(t) = y(t)$ and $x_2(t) = \dot{x}_1(t) = \dot{y}(t)$, the state variable representation is given by:

$$\dot{x}(t) = \mathrm{A}x(t) + bu(t) \tag{9.1}$$

$$y(t) = cx(t) + du(t) \tag{9.2}$$

where
$$\mathrm{A} = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix},\ b = \begin{bmatrix} 0 \\ \mathrm{K}_m \end{bmatrix},\ c = [1\ 0],\ d = 0 \text{ and } \mathrm{K}_m = 100 \tag{9.3}$$

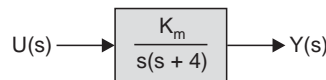



Fig. 9.1 Transfer function of a servomotor

Here $x_1(t)$ represents the angular position of the motor shaft and $x_2(t)$ represents its angular velocity, which can be sensed by a potentiometer and a tachogenerator, respectively. Suppose that, from the consideration of steady state performance, we set $\mathrm{K}_m = 100$ (vide Example 8.3 of Chapter 8). With a control law of $u(t) = -x_1(t)$, characteristic equation of the closed loop system becomes $|\,s\mathrm{I} - \mathrm{A}\,| = s^2 + 4s + 100 = 0$ with roots at

$$s_1 = -2.0000 + j9.7980,\ s_2 = -2.0000 - j\,9.7980$$

The closed loop system has damping ratio of 0.2 and the step response has overshoot of 53 percent and settling time of 1.96 sec. Suppose we want to increase the damping ratio to 0.530 and for meeting this requirement the closed loop poles are placed at $s = -5 \pm j8$ corresponding to the characteristic equation $s^2 + 10s + 89 = 0$.

In the state feedback design, the control signal input u is realized as linear combinations of all the states, that is

$$u(t) = -k_1 x_1(t) - k_2 x_2(t) = -K'x(t) \tag{9.4}$$

where the gain matrix K is a column vector and its transpose $K' = [k_1\ k_2]$ is a row.

The Equation (9.1) can then be written as

$$\dot{x}(t) = A_f x(t) \tag{9.5}$$

where

$$A_f = A - bK' = \begin{bmatrix} 0 & 1 \\ -100k_1 & -4-100k_2 \end{bmatrix} \tag{9.6}$$

The characteristic equation is given by

$$|\, sI - A_f \,| = 0 \tag{9.7}$$

which simplifies, for this example, to :

$$s^2 + (4 + 100\,k_2)s + 100\,k_1 = 0 \tag{9.8}$$

Let the desired location of the second order closed loop poles be designated as λ_1 and λ_2 such that the desired characteristic equation, denoted by $\alpha_c(s)$, is given by

$$\alpha_c(s) = (s - \lambda_1)(s - \lambda_2) = s^2 - (\lambda_1 + \lambda_2)s + \lambda_1 \lambda_2 \tag{9.9}$$

Equating the coefficients of similar powers of s in Equations (9.9) and (9.8) and solving for k_1 and k_2, we get

$$k_1 = -0.01\,\lambda_1 \lambda_2 \text{ and } k_2 = -0.01\,(4 + \lambda_1 + \lambda_2) \tag{9.10}$$

Thus we can find the gain matrix K such that the closed loop poles are located at any desired values λ_1 and λ_2.

With $\lambda_1 = -5 + j8$ and $\lambda_2 = -5 - j8$, the state feedback coefficients become

$$k_1 = 0.8900 \text{ and } k_2 = 0.0600 \tag{9.11}$$

In general, an nth order system is modeled by

$$\dot{x}(t) = Ax(t) + Bu(t) \tag{9.12}$$

and the control input $u(t)$ is generated as a linear combination of the states as

$$u(t) = -K'x(t) \tag{9.13}$$

where,

$$K' = [k_1\ k_2\ \ldots\ k_n] \tag{9.14}$$

Then Equation (9.12) can be written as

$$\dot{x}(t) = (A - BK')x(t) \tag{9.15}$$

We choose the desired pole locations

$$s_i = \lambda_i,\ i = 1 \text{ to } n \tag{9.16}$$

Then the closed loop system characteristic equation is

$$\alpha_c(s) = |\, sI - A + BK' \,| = (s - \lambda_1)(s - \lambda_2)\ldots(s - \lambda_n) \tag{9.17}$$

In this equation there are n unknowns, $k_1, k_2 \ldots k_n$ and n known coefficients in the right hand side of (9.17). We can solve for the unknown gains on the left hand side by equating coefficients of equal powers of s on the right hand side in Equation (9.17).

9.1.1 Ackerman's Formula

A more practical procedure for calculating the gain matrix K is the use of Ackerman's formula, which will be presented here without proof.

We begin with the plant model of (9.12). A matrix polynomial is formed using the coefficients of the desired characteristics polynomial

$$\alpha_c(s) = s^n + \alpha_{n-1} s^{n-1} + \ldots + \alpha_1 s + \alpha_o \qquad (9.18)$$

According to Cayley-Hamilton Theorem , *a matrix satisfies its own characteristic equation*, so by replacing s with the matrix A in the above polynomial we get (I is a $n \times n$ identity matrix).

$$\alpha_c\,(A) = A^n + \alpha_{n-1} A^{n-1} + \ldots\ \alpha_1 A + \alpha_o I \qquad (9.19)$$

Then Ackerman's formula for the gain matrix K is given by

$$K' = [0\ 0\ 0\ \ldots\ 0\ 1]\ [B\ AB\ \ldots\ A^{n-2}B\ \ A^{n-1}B]^{-1}\ \alpha_c\,(A) \qquad (9.20)$$

We note that $[B\ AB\ \ldots\ A^{n-2}B\ \ A^{n-1}B]$ is the controllability matrix and Ackerman's formula will produce a result if the system is controllable.

We shall now consider an example to apply the Ackerman's formula.

Example 9.1 The plant model of Equation (9.1) is given by

$$\dot{x}(t) = Ax(t) + bu(t) = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 100 \end{bmatrix} u(t)$$

and the desired characteristic polynomial is

$$\alpha_1(s) = (s - \lambda_1)(s - \lambda_2) = (s + 5 - j8)\ (s + 5 + j8) = s^2 + 10\ s + 89$$

Hence $$\alpha_c\,(A) = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix}^2 + 10 \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} + 89 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 89 & 6 \\ 0 & 65 \end{bmatrix}$$

Also $$Ab = \begin{bmatrix} 100 \\ -400 \end{bmatrix},\ \text{thus}\ [b\ Ab]^{-1} = \begin{bmatrix} 0 & 100 \\ 100 & -400 \end{bmatrix}^{-1} = \begin{bmatrix} 0.0400 & 0.0100 \\ 0.0100 & 0 \end{bmatrix}$$

Hence from (9.20),

$$K' = [0\ 1] * [b\ Ab]^{-1} * \alpha_c\,(A)$$

or, $$K' = [0.8900 \quad 0.0600]$$

These results are the same as those obtained in relation (9.11).

The MATLAB command for finding the state feedback coefficients by Ackerman formula is

K = *acker* (A, *b, p*)

where $p = [\lambda_1\ \ \lambda_2]$, is the desired closed loop poles, A is the system matrix, and b is the input matrix.

Ackerman's formula, in MATLAB environment, produces reliable results for systems of order 10 or less and works even with repeated assigned roots.

There is another MATLAB command *place* which is applicable even for multi-input-multi-output system with non-repeated poles that works well with systems of order higher than 10. The format for *place* is

[K, prec, message] = *place*(A, B, *p*)

where *prec* measures the number of accurate decimal digits in the desired non-repeated closed-loop poles p. In cases, due to ill conditioned coefficients in the characteristic equation, some nonzero closed-loop poles computed by the above MATLAB command is more than 10% off from the desired location, *message* contains a warning.

In the Example 9.1, the feedback coefficient k_1 can be realized by a ten turn potentiometer set to 0.890 while k_2 can be set by a Tachogenerator coupled to the output motor shaft and associated attenuator.

9.1.2 Guidelines for Placement of Closed Loop System Poles

The computation of the feedback gain is straight forward once the location of the closed loop poles are chosen. However, the choice of the poles are a bit tricky. We can choose the dominant pole pairs of the closed loop system rather easily from the consideration of transient response. But, the choice of other poles, especially those that affect high frequency response, are not that clear. However, a guideline is:

(*a*) *not to shift the open loop plant-poles without any specific objective like increasing the damping ratio*

(*b*) *not to shift the plant-poles which are close to plant zeros because that will cost plenty of control energy*

(*c*) *shift poles by the minimum amount that is absolutely necessary*

The final design should be accepted after assessing the initial condition response , step response, steady state error, gain margin and phase margin together with frequency response. State variable feedback design is very useful when used with other techniques like optimization (vide Chapter 11).

9.1.3 Linear Quadratic Regulator Problem

In a *Linear Quadratic Regulator* problem, the control strategy is to find the control law in Equation (9.13) which will minimize a performance index of the form

$$J = \int_0^{\infty} (x'Qx + u'Ru)dt$$

where Q is the $n \times n$ weight matrix for the states, R is an $m \times m$ weight matrix for m number of control inputs. For a single-input-single-output system, $m = 1$ and R is a scalar. The choice of R and Q are guided by the desire of the designer to give relative weights to the cost of control energy and system response and is normally done by trial and error. Usually, Q is a diagonal matrix. The thumb rules for selecting Q matrix are: (*i*) increasing the value of Q increases bandwidth, (*ii*) increasing the diagonal element q_{ii} in the weight matrix Q, increases damping when the ith state x_i is velocity-like, (*iii*) response of a state x_j can be made faster by increasing the diagonal entry q_{jj} larger in the weight matrix Q.

The MATLAB command that solves for the feedback matrix K that minimizes the above cost function J is :

[K, G, E] = *lqr* (A, B, Q, R), for continuous system

[K, G, E] = *dlqr* (A, B, Q, R), for discrete system

where K is the feedback matrix, G is the solution of matrix Riccati equation (vide Chapter 11) and E is the eigen value of the closed loop system E = *eig*(A-BK).

In the approach of *Linear Quadratic Regulator*, selection of a larger weight matrix R relative to Q indirectly avoids picking closed loop poles that increases cost of control energy.

For the continuous system in Example 9.1 with $A = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix}$, $b = \begin{bmatrix} 0 \\ 100 \end{bmatrix}$

and (*i*) $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.01 \end{bmatrix}$, R = 10 ; (*ii*) $Q = \begin{bmatrix} 1 & 0 \\ 0 & 0.1 \end{bmatrix}$, R = 10 ; (*iii*) $Q = \begin{bmatrix} 10 & 0 \\ 0 & 0.1 \end{bmatrix}$, R = 10

the respective feedback matrices and the corresponding eigen values that minimizes the cost J are found by the MATLAB command *lqr* as :

(*i*) $k' = [0.3162 \quad 0.0545]$, $E = [-4.7235 + j3.0515 \quad -4.7235 - j3.0515]$

(*ii*) $k' = [0.3162 \quad 0.0939]$, $E = [-3.0625 \quad -10.3257]$

(*iii*) $k' = [1.0000 \quad 0.1378]$, $E = [-8.8882 + j4.5826 \quad -8.8882 - j4.5826]$

The value of q_{22} has been increased from 0.01 in (*i*) to 0.1 in (*ii*), which has pronounced effect on the damping. Also, the effects of increasing the elements of Q by a factor of 10 [compare Q in (*i*) and (*iii*)] are found to produce larger bandwidth associated with increased value of the imaginary component in the eigen value.

The pole assignment design technique assumes that all the plant states are available for measurement and feedback. However, this assumption is difficult to satisfy in many situations.

9.2 STATE ESTIMATION

In general, the measurement of all the states is impractical, if not impossible, in all but the simplest of systems. A technique of estimating the states of a plant is presented in this section using information that is available from the plant. The system that estimates the states of another system is generally called an observer or state estimator.

Suppose that the plant is described by the following equation

$$\dot{x}(t) = Ax(t) + Bu(t) \qquad (9.21)$$
$$y(t) = Cx(t)$$

where $y(t)$ are the plant signals that can be measured.

Hence in Equation (9.21) we know matrices A, B and C and the signals $y(t)$ and $u(t)$. The control inputs $u(t)$ are known since the designer generates them. The problem of the estimator design can be depicted as shown in Fig. 9.2, where the estimator is a set of differential equations to be solved by an analog computer. These states of the system to be observed are $x(t)$, the states of the estimator is $q(t)$ and we desire that the state $q(t)$ be approximately equal to $x(t)$. Since the estimator will be implemented on a computer, the signals $q(t)$ are then available for feedback design.

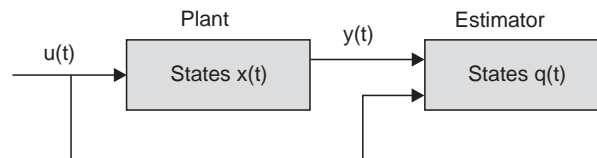



Fig. 9.2 State estimation when output is available for measurement

The estimator design criterion to be used is that the transfer function matrix from the input $u(t)$ to the estimator state $q_i(t)$ be equal to that from the input $u(t)$ to the system state $x_i(t)$ for $i = 1, 2, \ldots\ldots n$.

Now taking the Laplace transform of both sides of Equation (9.21) with zero initial condition, we have

$$sX(s) = AX(s) + BU(s) \qquad (9.22)$$

or

$$X(s) = (sI - A)^{-1} BU(s) \qquad (9.23)$$

Since the estimator has two inputs $y(t)$ and $u(t)$, we write the estimator state equation as

$$\dot{q}(t) = Fq(t) + Gy(t) + Hu(t) \tag{9.24}$$

where the matrices F, G and H are unknown and are to be found out.

Taking Laplace transform of both sides of Equation (9.24) with zero initial conditions and solving for Q(s), we get

$$Q(s) = (sI - F)^{-1} [GY(s) + HU(s)] \tag{9.25}$$

From (9.21), $$Y(s) = CX(s) \tag{9.26}$$

Next substitute (9.26) and (9.23) in (9.25)

$$Q(s) = (sI - F)^{-1} [GC (sI - A)^{-1} B + H] U(s) \tag{9.27}$$

Recall that the design criterion is that the transfer function matrix from U(s) to Q(s) be the same as that from U(s) to X(s). Thus from (9.23)

$$Q(s) = (sI - A)^{-1} BU(s) \quad \text{must be satisfied.} \tag{9.28}$$

Then from (9.28) and (9.27), we have,

$$(sI - A)^{-1} B = (sI - F)^{-1} GC (sI - A)^{-1} B + (sI - F)^{-1} H \tag{9.29}$$

or, $$[I - (sI - F)^{-1} GC] (sI - A)^{-1} B = (sI - F)^{-1} H$$

Pre-multiplying both sides by (sI – F), we get

$$[(sI - F) - GC] (sI - A)^{-1} B = H \tag{9.30}$$

Solving for $(sI - A)^{-1}$ B, we have

$$(sI - A)^{-1} B = [sI - (F + GC)]^{-1} H \tag{9.31}$$

This equation is satisfied if we choose H equal to B and

$$A = F + GC \tag{9.32}$$

Therefore, from (9.24) and (9.32), we write the estimator state equation

$$\dot{q}(t) = (A - GC)\, q(t) + Gy(t) + Bu(t). \tag{9.33}$$

and the design criterion (9.28) is satisfied.

Define the error vector $e(t)$ as

$$e(t) = x(t) - q(t) \tag{9.34}$$

Then taking derivative of both sides of Equation (9.34) and using (9.21) and (9.33), we have

$$\dot{e}(t) = \dot{x}(t) - \dot{q}(t)$$

$$= Ax(t) + Bu(t) - (A - GC)q(t) - GCx(t) - Bu(t) \tag{9.35}$$

or $$\dot{e}(t) = (A - GC)\, [x(t) - q(t)] = [A - GC]\, e(t) \tag{9.36}$$

Hence the error dynamics have the same characteristics equation as that of the estimator in Equation (9.33).

9.2.1 Sources of Error in State Estimation

The above scheme for state estimation shown in Fig. 9.2 suffers from numerous sources of error listed below.

(1) Model of the physical system may not be exact

(2) Choice of initial conditions for the estimator can not be made identical to that of the plant

(3) Plant disturbance and measurement noise can not be simulated exactly

The effect of the plant disturbances and measurement noise can be studied by considering them as additional inputs to the plant model as shown below

$$\dot{x}(t) = Ax(t) + Bu(t) + B_1 w(t) \quad (9.37)$$

and $$y = x(t) + v(t)\,;$$

where $w(t)$ is the plant disturbance and $v(t)$ is the measurement noise. The error in state estimation, therefore, is given by

$$\dot{e}(t) = (A - GC)\, e(t) + B_1\, w(t) - Gv(t)$$

9.2.2 Computation of the Observer Parameters

Consider now the observer equation

$$\dot{q}(t) = (A - GC)\, q(t) + Gy(t) + Bu(t) \quad (9.38)$$

All matrices in the observer equation (9.38) are determined by the plant equation with the exception of the matrix G; however, the estimator design criteria (9.32) is satisfied without any constraint on the choice of the matrix G. But, we note from Equation (9.36) that the error dynamics is dependent on G and this may be used as a guiding factor for choosing G. In view of the fact that the determinants of a matrix A and its transpose A′ remains invariant, the characteristic equation of the estimator may be written as

$$\alpha_c(s) = |\, sI - (A - GC) | = |\, sI - (A' - C'G')\, | = s^n + \alpha_{n-1}\, s^{n-1} + \ldots\ldots + \alpha_1 s + \alpha_o \quad (9.39)$$

By comparing the characteristic equation $\alpha_c(s) = |\, sI - (A - BK') |$ in (9.17) with Equation (9.39), we note the following association

$$A \to A'\,,\ K' \to G'\,,\ \ B \to C' \quad (9.40a)$$

Hence G can be computed from the Ackerman's equation (9.20) as

$$G' = [0\ 0\ \ 0 \ldots 1\,]\ [C'\ A'C' \ldots A'^{n-1}C'\,]^{-1}\ \alpha_c\,(A') \quad (9.40b)$$

or $$G = \alpha_c\,(A) \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (9.40c)$$

Thus we see that if the system is **observable**. we can design the estimator, once we decide on an appropriate characteristic equation (9.39). **An approach usually suggested for choosing α_e (s) is to make the estimator response 2 to 4 times faster than that of the closed loop control system**. The fastest time constant of the closed loop system determined from the system characteristic equation

$$|\, sI - (A - BK')\, | = 0 \quad (9.41)$$

is calculated first. The time constants of the estimator is then set at a value equal to from **one fourth to one-half** of this fastest time constant, whenever feasible.

The choice of G can be approached from another viewpoint by writing the estimator state equation as

$$\dot{q}(t) = Aq(t) + G[y(t) - Cq(t)] + Bu(t) \quad (9.42)$$

The Equation (9.42) is represented by a simulation diagram of the plant and estimator as shown in Fig. 9.3, where the measurement noise, $v(t)$ and the plant disturbance, $w(t)$ are shown explicitly.

Using the Equation (9.37), the estimator dynamics in Equation (9.42) can be written as :

$$\dot{q}(t) = Aq(t) + G[y(t) + v(t) - Cq(t)] + Bu(t) + B_1 w(t)$$

or,
$$\dot{q}(t) = Aq(t) + G[y(t) - Cq(t)] + Bu(t) + B_1 w(t) + Gv(t) \quad (9.43)$$

In the above equation the sensor noise $v(t)$ is multiplied by G while the plant disturbance is not affected by G.

If $x(t)$ and $q(t)$, in Fig. 9.3, are nearly equal, then there is little contribution through the feed back path containing G. Hence, $q(t)$ is mainly dependent on $u(t)$. However, if the effects of disturbances $w(t)$ on the plant cause $q(t)$ to differ significantly from $x(t)$, the contribution of G becomes considerable. However, if the measurement noise $v(t)$ is appreciable, G should be made small so as to keep $Gv(t)$ to a relatively small value. On the other hand, if G is made too small, the effect of measurement noise will be reduced, but the estimator's dynamic response will be too slow to attenuate the plant disturbance $w(t)$ properly. A low gain estimator will also fail to minimize any error arising out of plant-modeling error. Hence for practical systems, the best option for choosing G is through the simulation of the system to attain a balance caused by the effects of plant disturbances and the measurement noise.

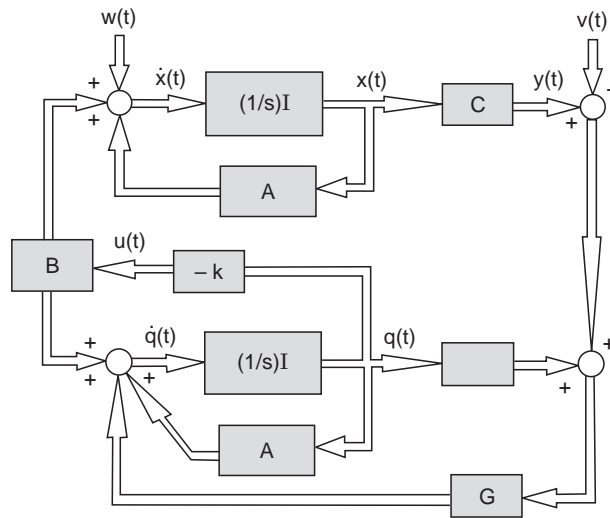



Fig. 9.3 Block diagram of estimator

Simulation should be run for various values of G obtained from different choices of $\alpha_c(s)$, with the final choice of G resulting from the acceptable system response. The optimal solution to attain the balance between plant disturbance intensity P_w and measurement noise intensity N_v can be found through Kalman estimator. The detail treatment of Kalman estimator will not be undertaken here. However, G along with the Kalman estimator gain *Kest,* can be calculated using MATLAB command as :

$$[\text{Kest}, G, P_{ss}] = \textit{Kalman}\ (\text{sys}, P_w, N_v)$$

where P_{ss} is a steady state error measure and *sys* is the system matrices in the state equation.

An example of the design of an estimator will now be presented.

Example 9.2 We will design an estimator for the system of Example 9.1

$$\dot{x}(t) = Ax(t) + Bu(t) = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 100 \end{bmatrix} u(t)$$

$$y(t) = [1 \quad 0]\, x(t)$$

With the gain matrix K′ = [0.8900 0.0600], the closed loop system characteristic equation is given by

$\alpha_c(s) = s^2 + 10\,s + 89 = 0$, with desired poles located at $s_{1,2} = -5 \pm j8$

The time constant of the roots of this equation is $\tau = 1/\delta\omega_n = 1/5 = 0.2$ sec . We choose the time constant of the estimator to be $\tau/2 = 0.1$ sec. We also choose the estimator roots to be real and equal. Hence the root locations are

$$s_{1,2} = -1/0.1 = -10$$

The estimator characteristic equation is then,

$$\alpha_e(s) = (s + 10)^2 = s^2 + 20\,s + 100 = 0$$

The matrix G is given by $\quad G = \alpha_e(A)\begin{bmatrix} C \\ CA \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Now, $\quad \alpha_e(A) = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix}^2 + 20\begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} + 100\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 100 & 16 \\ 0 & 36 \end{bmatrix}$

$$C = [1\ 0],\ CA = [0\ 1],\ \text{Hence,}\ \begin{bmatrix} C \\ CA \end{bmatrix}^{-1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Then $\quad G = \begin{bmatrix} 16 \\ 36 \end{bmatrix}$ and $F = A - GC = \begin{bmatrix} -16 & 1 \\ -36 & -4 \end{bmatrix}$

From (9.33), the estimator state equation is:

$$\dot{q}(t) = F\, q(t) + Gy(t) + Bu(t)$$

Substituting the values of F, G and B we get,

$$\dot{q}(t) = \begin{bmatrix} -16 & 1 \\ -36 & -4 \end{bmatrix} q(t) + \begin{bmatrix} 16 \\ 36 \end{bmatrix} y(t) + \begin{bmatrix} 0 \\ 100 \end{bmatrix} u(t)$$

But $u(t) = -K'\, q(t) = -0.8900\, q_1(t) - 0.0600\, q_2(t)$

Therefore, the estimator dynamics are

$$\dot{q}(t) = \begin{bmatrix} -16 & 1 \\ -125 & -10 \end{bmatrix} q(t) + \begin{bmatrix} 16 \\ 36 \end{bmatrix} y(t)$$

Initial condition responses for an estimator-based control system are given in Fig. 9.4. Two responses are given one with (*a*) $q(0) = x(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and (*b*) the other with $q(0) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $x(0) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$.

The choice of the initial conditions will simulate the effect of the observer-based control system where the initial state of the plant is not known completely and hence cannot be made equal to the initial states of the estimator. With equal initial values $q(0)$ and $x(0)$, the response of the estimator-based control system will be identical to that of the full-state feedback system of Example 9.1. In Fig. 9.5 the initial condition responses of the estimator-based control system and the direct-state-feedback system are shown with a step disturbance on the plant. The disturbance enters state model (9.37) with

$$B_1 = \begin{bmatrix} 0 \\ 10 \end{bmatrix}$$

From Fig. 9.5, we find that the response of the estimator-based control system differs significantly from that of the direct-state feedback system, since the estimator-based feedback does not take disturbances into account. The cost of control energy is proportional to $u(t)^2$, where $u(t)$ is given by $u(t) = K'x(t)$. The control energy should not be too high while implementing control law for pole assignment design.

9.3 EQUIVALENT FREQUENCY-DOMAIN COMPENSATOR

Just to satisfy our curiosity, let us relate the pole assignment design to frequency-domain design of an equivalent compensator. The estimator equations are

$$\dot{q}(t) = (A - GC)\, q(t) + Gy(t) + Bu(t) \tag{9.44}$$

and the control law is given by $u(t) = - K'q(t)$ (9.45)

substitution for the equation for $u(t)$ into the estimator equation yields

$$\dot{q}(t) = (A - GC - BK')\, q(t) + Gy(t) \tag{9.46}$$

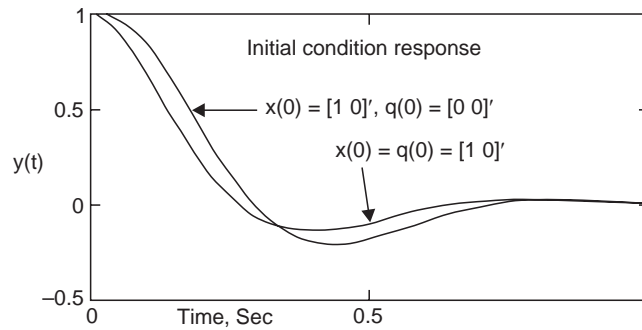



Fig. 9.4 Initial condition response for Example 9.2

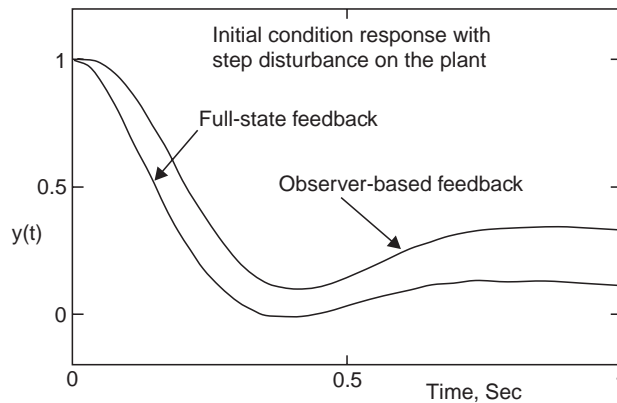



Fig. 9.5 Initial condition response with step disturbance for Example 9.2

Taking the Laplace transform of both sides of this equation with zero initial conditions and solving for $Q(s)$ yields

$$Q(s) = (sI - A + BK' + GC)^{-1}\, GY(s) \tag{9.47}$$

Substituting this equation into that for U(s) results in the relationship

$$U(s) = - K'(sI - A + BK' + GC)^{-1}\, GY(s) \tag{9.48}$$

Hence we can consider the control-estimator combination to be an equivalent frequency-domain compensator with transfer function

$$D_{ce}(s) = -\frac{U(s)}{Y(s)} = K' (sI - A + BK' + GC)^{-1} G \tag{9.49}$$

and the closed loop system can be represented by a structure shown in Fig. 9.6(*a*).

Note that, for this controller, – Y(*s*) is the input and U(*s*) is the output. In Fig. 9.6(a) we show the control system as an equivalent unity gain feedback system, with the system input equal to zero. We can also view the system with no input shown, as in Fig. 9.6(b). The system characteristic equation can be expressed as

$$1 + D_{ce}(s)G(s) = 0 \tag{9.50}$$

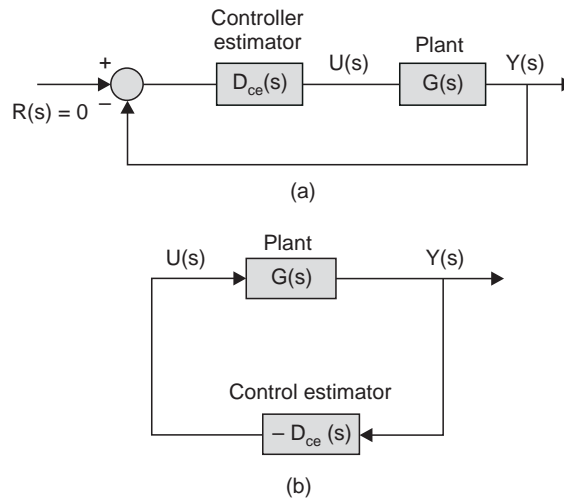



Fig. 9.6 Equivalent frequency-domain controller realization of pole placement design

9.4 COMBINED PLANT AND OBSERVER DYNAMICS OF THE CLOSED LOOP SYSTEM

We will now derive the system characteristic equation for the observer-based control system for which we use the error variables of (9.34)

$$e(t) = x(t) - q(t) \tag{9.51}$$

The plant state equations of (9.21) can then be expressed as

$$\dot{x}(t) = Ax(t) - BK' q(t) = (A - BK') x(t) + BK'e(t) \tag{9.52}$$

The state equations for the error variables are

$$\dot{e}(t) = (A - GC) e(t) \tag{9.53}$$

We can adjoin the variables of (9.52) and (9.53) into a single state vector, with the resulting equations

$$\begin{bmatrix} \dot{x}(t) \\ \dot{e}(t) \end{bmatrix} = \begin{bmatrix} A - BK' & BK' \\ 0 & A - GC \end{bmatrix} \begin{bmatrix} x(t) \\ e(t) \end{bmatrix} \tag{9.54}$$

with the states of the observer a linear combination of *x*(*t*) and *e*(*t*) given in (9.51) . Thus the characteristic equation of the state equations (9.54) is also the closed-loop system characteristic equation. This equation is found to be

Chapter 9

$$| sI - A + BK' | \, | sI - A + GC | = \alpha_c(s) \, \alpha_e(s) = 0 \tag{9.55}$$

We see that the roots of the characteristic equation of the closed loop system are the roots obtained by pole-placement design plus those of the observer. **The pole-placement design is, thus, independent of the observer dynamics.**

A state model of the closed loop system will now be derived. From relation (9.21) and using $q(t)$ as the estimated value of $x(t)$ in (9.13), we can write

$$\dot{x}(t) = A\, x(t) - BK'\, q(t)$$

and from (9.33)

$$\dot{q}(t) = GC\, x(t) + (A - GC - BK')\, q(t)$$

We adjoin the foregoing equations to form the closed-loop state model

$$\begin{bmatrix} \dot{x}(t) \\ \dot{q}(t) \end{bmatrix} = \begin{bmatrix} A & -BK' \\ GC & A - GC - BK' \end{bmatrix} \begin{bmatrix} x(t) \\ q(t) \end{bmatrix} \tag{9.56}$$

Example 9.3 The closed-loop state matrix (9.56) will now be calculated for the system of Example 9.2. We note from Example 9.2

$$BK' = \begin{bmatrix} 0 \\ 100 \end{bmatrix} [0.8900 \; 0.0600] = \begin{bmatrix} 0 & 0 \\ 89 & 6 \end{bmatrix}$$

$$GC = \begin{bmatrix} 16 \\ 36 \end{bmatrix} [1 \; 0] = \begin{bmatrix} 16 & 0 \\ 36 & 0 \end{bmatrix}$$

From Example 9.2, we get $A - GC - BK' = \begin{bmatrix} -16 & 1 \\ -125 & -10 \end{bmatrix}$

The closed-loop system matrix, from (9.56), is then

$$\begin{bmatrix} A & -BK' \\ GC & A - GC - BK' \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -4 & -89 & -6 \\ 16 & 0 & -16 & 1 \\ 36 & 0 & -125 & -10 \end{bmatrix}$$

9.5 INCORPORATION OF A REFERENCE INPUT

When the command input $u(k)$ is realized as a linear combination of the states to implement the control laws, the resulting closed loop system is regulator type where the design objective is to drive all the states to equilibrium state following an initial condition response or plant disturbance. However, we can keep provision for a reference input in a state feedback design approach. This can be done by taking the difference of the output y with reference input as shown in Fig. 9.7. In this configuration, the command input is directly applied to the observer, followed by the plant. So the estimator characteristic is used to generate the control law. In an alternate arrangement, the reference input is impressed to the plant as well as to the estimator (vide Fig. 9.8).The estimator and the plant, therefore, respond to the command input and the control law in an identical fashion, avoiding the estimator error characteristic. So this is a preferred arrangement. The input matrix B_r in Fig. 9.8 should be chosen to eliminate any steady state error. If x_{ss} and u_{ss} are the steady state values of the system states and control u, B_r should be selected to bring the states to equilibrium and steady state output of 1 for unity command.

The state equations is recast as $\begin{bmatrix} \dot{x}(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x(t) \\ u(t) \end{bmatrix}$

which, in the steady state condition becomes $\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x_{ss}(t) \\ u_{ss}(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$,

so that the x_{ss} and u_{ss} can be solved from $\begin{bmatrix} x_{ss}(t) \\ u_{ss}(t) \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ (9.57)

Again, we note from the Fig. 9.8, that

$$u_{ss} = rB_r - Kq_{ss} = B_r - Kq_{ss}\ , \text{ for } r = 1$$

So, the matrix B_r, is found as

$$B_r = Kq_{ss} + u_{ss} \quad (9.58)$$

where q_{ss} (= x_{ss}) and u_{ss} are given by (9.57).

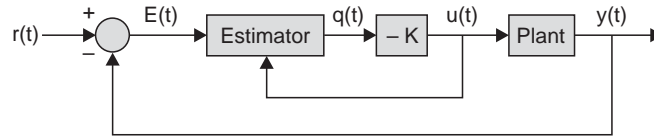



Fig. 9.7 Estimator in the forward path

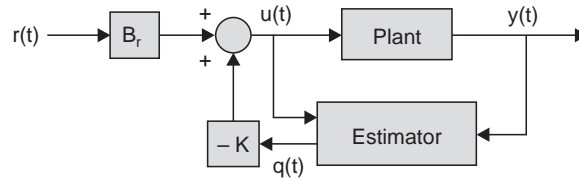



Fig. 9.8 Estimator in the feedback path

9.6 REDUCED-ORDER OBSERVER

In a full observer design approach, for generating the control law, all the states of the plant were synthesized through computer simulation, irrespective of their availability from the sensor output. Some of the undesirable effects of full state feedback, like the plant disturbances, can be minimized by using as many states of the system as are available for state feedback and the rest being estimated through computer simulation.

In order to develop the design equations for the reduced order observer, we partition the state vector as

$$x(t) = \begin{bmatrix} x_a(t) \\ x_b(t) \end{bmatrix}$$

where $x_a(t)$ are the states to be measured and $x_b(t)$ are the states to be estimated. Then the plant state equation of (9.21) can be partitioned as

$$\begin{bmatrix} \dot{x}_a(t) \\ \dot{x}_b(t) \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} \begin{bmatrix} x_a(t) \\ x_b(t) \end{bmatrix} + \begin{bmatrix} B_a \\ B_b \end{bmatrix} u(t) \quad (9.59)$$

$$y(t) = [1\ 0] \begin{bmatrix} x_a(t) \\ x_b(t) \end{bmatrix}$$

The equations for the measured states can be written as

$$\dot{x}_a(t) = A_{aa}x_a(t) + A_{ab}\, x_b(t) + B_a u(t)$$

Collecting all the known terms and transferring them on the left side of the equation, we have

$$\dot{x}_a(t) - A_{aa}\, x_a(t) - B_a\, u(t) = A_{ab}\, x_b(t) \tag{9.60}$$

For the reduced order observer, we consider the left side to be the "known measurements" and the term $A_{ba}\, x_a(t) + B_b u(t)$ is then considered to be the "known inputs" .

We now compare the state equations for the full observer to those for the reduced-order observer.

$$\dot{x}(t) = Ax(t) + Bu(t)$$

$$\dot{x}_b(t) = A_{bb}\, x_b(t) + [A_{ba}\, x_a(t) + B_b\, u(t)] \tag{9.61}$$

and $$y(t) = Cx(t)$$

$$\dot{x}_a(t) - A_{aa}\, x_a(t) - B_a\, u(t) = A_{ab}\, x_b(t) \tag{9.62}$$

We then obtain the reduced-order observer equations by making the following substitutions into the full-order observer equations (9.33)

$$\begin{aligned} x(t) &\leftarrow x_b(t) \\ A &\leftarrow A_{bb} \\ Bu(t) &\leftarrow A_{ba}\, x_a(t) + B_b u(t) \\ y(t) &\leftarrow \dot{x}_a(t) - A_{aa}\, x_a(t) - B_a u(t) \\ C &\leftarrow A_{ab} \end{aligned}$$

If we make these substitutions into (9.33) and designating the estimator gain for the **reduced order as $\mathbf{G_r}$**, we obtain the equation

$$\dot{q}_b(t) = (A_{bb} - G_r A_{ab})\, q_b(t) + G_r[\dot{x}_a(t) - A_{aa}\, x_a(t) - B_a u(t)] + A_{ba}\, x_a(t) + B_b u(t) \tag{9.63}$$

From (9.59) $$y(t) = x_a(t) \tag{9.64}$$

Then (9.63) can be written as

$$\dot{q}_b(t) = (A_{bb} - G_r A_{ab})\, q_b(t) + G_r\, \dot{y}(t) + (A_{ba} - G_r A_{aa})\, y(t) + (B_b - G_r B_a)\, u(t) \tag{9.65}$$

The following points should be made about the reduced order observer. The observer characteristic equation is

$$\alpha_e(s) = |\ sI - A_{bb} + G_r\, A_{ab}\ | = 0 \tag{9.66}$$

and G_r is computed in the same manner from (9.40) used for computing G for full-order observer. For the case of a single measurement [*i.e.*, $y(t)$ is $x_1(t)$], Ackerman's formula is given by

$$G_r = \alpha_e\,(A_{bb}) \begin{bmatrix} A_{ab} \\ A_{ab}\, A_{bb} \\ \\ A_{ab}\, A_{bb}^{n-2} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{9.67}$$

Note also that, in (9.65), measurements of $\dot{y}(t)$ is required to estimate $\dot{q}_b(t)$. However, for the full order observer in (9.33), $\dot{q}(t)$ is estimated using the measurements of $y(t)$ only.

Assuming $y(t) = x_1(t)$, and $u(t)$ is given by partitioned $u(t)$ as :

$$u(t) = -[k_1 \ \ k_b]\ [y(t)\ q_b(t)]'$$
$$= -k_1\, y(t) - k_b q_b(t)$$

the equivalent frequency-domain compensator $D_{ce}(s)$ is given by

$$D_{ce}(s) = \frac{-U(s)}{Y(s)}$$
$$= k_1 + k_b\ [sI - A_{bb} + G_r A_{ab} + (B_b - G_r\ B_a)\ k_b]^{-1}$$
$$[G_r s + \{A_{ba} - G_r A_{aa} - k_1(B_b - G_r B_a)\}] \quad (9.68)$$

Example 9.4 We will again consider the design of the system of Example 9.2. The plant model is given by

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 100 \end{bmatrix} u(t), \ \ y(t) = [1 \ \ 0]\ x(t)$$

Thus we are measuring position, $x_1(t)$ and will estimate velocity $x_2(t)$. The desired closed-loop system characteristic equation was chosen to be

$$\alpha_c(s) = s^2 + 10s + 89 = 0.$$

We will make the same choice for the estimator characteristic equation roots $s = -10$ as in Example 9.2. However, since the reduced-order observer here is first order, the characteristic equation is given by

$$\alpha_e(s) = s + 10 = 0$$

From the plant equation above,

$$A_{aa} = 0, \quad A_{ab} = 1, \quad B_a = 0$$
$$A_{ba} = 0, \quad A_{bb} = -4, \quad B_b = 100$$

Thus we have all the terms required for Ackermann's formula (9.67)

$$\alpha_e(s) = s + 10, \text{ Thus } \ \alpha_e(A_{bb}) = -4 + 10 = 6$$
$$G_r = \alpha_e\ (A_{bb})\ (A_{ab})^{-1}\ [1] = 6\ (A_{ab})^{-1}\ [1] = 6$$

From (9.65), the observer equation is given by

$$\dot{q}(t) = (A_{bb} - G_r A_{ab})\ q(t) + G_r\ \dot{y}(t) + (A_{ba} - G_r A_{aa})\ y(t) + (B_b - G_r B_a)\ u(t)$$

or
$$\dot{q}(t) = -10q(t) + 6\ \dot{y}(t) + 100\ u(t)$$

Here $q(t)$ is the estimate of velocity $x_2(t)$.

From Example 9.2 the control law is given by

$$u(t) = -0.89\ x_1(t) - 0.06\ x_2(t)$$

which is implemented as
$$u(t) = -0.89\ y(t) - 0.06\ q(t)$$

Hence we can write the observer equation as

$$\dot{q}(t) = -10q(t) + 6\dot{y}(t) + 100\ [-0.89\ y(t) - 0.06q(t)] = -16q(t) + 6\dot{y}(t) - 89\ y(t)$$

The control system is then implemented as follows. A measurement of y(t) is made at time t. The observer state is calculated from $\dot{q}(t) = -16\ q(t) + 6\ \dot{y}(t) - 89\ y(t)$.

The initial condition response, obtained by simulation is approximately the same as that obtained using full state feedback. In addition, the effects of the disturbance input are less than those for the full-order observer system.

Example 9.5 We will now calculate the transfer function of the equivalent controller estimator for the system of Example 9.4. From Example 9.4, $G_r = 6$ and

$$A_{aa} = 0, A_{ab} = 1, B_a = 0$$
$$A_{ba} = 0, A_{bb} = -4, B_b = 100$$
$$k_1 = 0.8900\ k_b = 0.0600$$
$$D_{ce}(s) = \frac{-U(s)}{Y(s)}$$
$$= k_1 + k_b\ [sI - A_{bb} + G_r\ A_{ab} + (B_b - G_r\ B_a)\ k_b]^{-1}$$
$$[G_r s + \{A_{ba} - G_r A_{aa} - k_1(B_b - G_r B_a)\}]$$
$$= 1.25\ (s + 7.168)/(s + 16)$$

The equivalent transfer function of the compensator is that of a lead compensator, since, the zero is closer to the origin in the s-plane than the pole, which is expected of a compensator improving the transient response.

9.7 SOME GUIDELINES FOR SELECTING CLOSED LOOP POLES IN POLE ASSIGNMENT DESIGN

In pole placement design the choice of closed loop poles should be such that it meets the transient response specifications, which, of course, is not an easy task except in trivial cases. In order to keep the control energy and hence its size and cost to the bare minimum that is absolutely necessary for a given situation, the poles corresponding to the natural modes of response of the plant should not be shifted too much than is absolutely necessary. For example, in lightly damped open loop system, emphasis should be given on increasing the damping ratio, not on the frequency, this will help to keep the control energy to a low value.

In optimal regulator problem, the location of poles are selected to keep the square of the cost function of the form

$$J = \int_{t=0}^{\infty} [p_{ii} e_i^2(t) + r_{ii}\ u_i^2(t)\ dt \tag{9.69}$$

to minimum, where $e(t)$ is the error of the state from the desired value and u_i^2 is proportional to control energy and p_{ii} and r_{ii} are, respectively, the weights associated to the error and control energy. The designer varies the relative weights between the state errors and the control to meet the system specifications with minimum control.

The design procedure of estimator-error pole selection is similar to pole placement design, however, the choice of poles are guided by different considerations. A small time constant for the estimator pole does not increase energy-cost, because estimators are realized by analog computer simulation. But fast estimator poles are responsible for increased sensitivity between sensor errors and estimator errors. The main objective for pole selection of estimator-error design is to minimize the plant disturbance and sensor noise. The estimator time constants are made smaller so that the system response is dominated by controller poles. If very small estimator time constant increases the estimator error due to sensor noise, it can be increased to more than half of the controller time constants. This choice of estimator pole, however, may influence the overall system response and the controller design and estimator design becomes interdependent.

In optimal estimator design, the choice of estimator-error poles are dependent on the ratio of plant model errors and sensor noise errors. For a plant with accurate model parameters and small disturbances but with large sensor noise, the optimal estimator error is found by

setting a low estimator gain which produces slow response. On the other hand, a plant with large modeling error, but with reliable sensor output, a large estimator gain is selected for correcting the plant disturbance quickly.

MATLAB SCRIPTS

```
%Script_Example_9.1
% Computation of feedback coefficients
A=[0 1;0 -4];      %system matrices refer to equation 9.3
b=[0;100];
c=[1 0];
Ab=A*b;            % type >> help ctrb
co=[b, Ab];        % Alternatively, find controllability matrix co=ctrb(A, b);
c2=inv(co);        % compute inverse of co, c2=inv(co)
alpha1=A^2+10*A+89*eye(2,2);    % Refer to equation (9.19) with n = 2
k=[0 1]*c2*alpha1 ;             % Refer to equation (9.20)
bk=b*k
Af=A-b*k
p=[-5+8i -5-8i];
 K_acker = acker(A, b, p)       % Feedback coeff computed by Acker formula
```

```
% Script_Example_9.2_9.3
% Observer-based controller
A=[0 1;0 -4];          % A, b, c are system matrices
b=[0;100];
c=[1 0];
p=[-5+8i -5-8i];       % P = desired pole location in the s-domain
k=acker(A, b, p)       % Feedback matrix k needed for desired close loop poles
bk=b*k
Af=A-bk;               % closed loop system matrix
p1=[-10 -10];          % Location of observer poles
G=acker(A′,c′,p1)′     % Observer gain, Refer to association in relation (9.40a)
Gc=G*c
F=A-G*c                % Observer matrix
Afo=F-b*k              % Closed loop system matrix realised with full observer
%Finding initial condition response for siso with diff initial conditions on x and q
Acomb=[A -bk;Gc Afo] % Combined system and observer matrix, vide equation (9.56);
b=[0;1;0;0];           % Input matrix for combined plant and observer dynamics
c1=[1 0 0 0];          % Output matrix for the combined plant and observer
d=[0];
sys=ss(Acomb,b,c1,d);  % Sys is the combined plant and estimator dynamics
T=0:0.01:1;
x0=[1;0;1;0];          % Equal initial values for plant states and estimator states
x00=[1;0;0;0];         % different initial values for plant states and estimator states
g1=initial(sys,x0,T);  % Response to initial condition x0
g2=initial(sys,x00,T); % Response to initial condition x00
plot(T,g1,'-',T, g2,'-')
```

```
% Script_Fig9.5
% Observer-based controller
A=[0 1;0 -4];            % A, b, c are system matrices
b=[0;100];
c=[1 0];
p=[-5+8i -5-8i];         % P = desired pole location in the s-domain
k=acker(A, b, p)         % Feedback matrix k needed for desired close loop poles
bk=b*k
Af=A-bk;                 % closed loop system matrix
p1=[-10 -10];            % Location of observer poles
G=acker(A′,c′,p1)′       % Observer gain, Refer to association in relation (9.40a)
Gc=G*c
F=A-G*c                  % Observer matrix
Afo=F-b*k                % Closed loop system matrix realised with full observer
%Finding initial condition response for siso with diff initial conditions on x and q
Acomb=[A -bk;Gc Afo] % Combined system and observer matrix, vide equation (9.56);
b1=[0; 10];              % Input matrix for plant disturbance
B1=[0;10;0;0];           % Input matrix for combined plant and estimator
c1=[1 0 0 0];
d=[0];
sys1=ss(Af,b1,c,d);         % Closed loop plant with disturbance input
sys2=ss(Acomb,B1,c1,d);   % closed loop plant combined with estimator
T=0:0.01:1;
x0=[1;0];
x00=[1;0;0;0];
 g1=initial(sys1,x0,T);    % Initial condition response of closed loop plant
 g2=step(sys1,T);          % Closed loop plant response to step disturbance
 g12=g1+g2;                % Initial condition response in presence of plant disturbance
 g3=initial(sys2,x00,T);   % Initial condition response of combined feedback system
 g4=step(sys2,T);          % Response of combined system to step disturbance
 g34=g3+g4;                % Initial condition response of combined feedback system
 plot(T,g12,'-',T, g34,'-')  % Initial condition response of combined system in
                            % presence of planrt disturbance
```

REVIEW EXERCISE

RE9.1 A field controlled dc motor incorporating an amplifier with gain A in the field circuit has transfer function given by :

$$G(s) = \frac{K_1}{s(s\tau_f + 1)(s\tau_m + 1)} = \frac{K}{s(s+5)(s+1)}$$

where $\tau_f = R_f / L_f = 0.2$ sec and $\tau_m = J/B = 1$ sec are the time constants of the field circuit and motor respectively and $K_1 = AK_T / R_f B$. Find the feedback coefficients to meet the specifications of the closed loop position control system such that the (1)damping ratio of the dominant roots is 0.625 and (2) settling time, within 2% of the final value, is less than 1 sec. In addition to meeting the specifications in the transient state, the steady state performance demands that $K_v > 200$.

Solution : Since δ = 0.625 and from relation (4.18) of chapter 4, $\tau_s = 4/(\delta\omega_n) < 1$, a possible choice of the dominant roots is $s_{1,2} = -4 \pm j5$. The absolute value of the third root should be greater than the real part of the dominant root *i.e.,* > 4. Let us place it at – 5 and set K = 205 to meet the specification on $K_v = K$. Following the procedure outlined in sec A.9.1 of Appendix A, the state variable representation of the system with physical variables of angular position, angular velocity and field current as state variables (vide Equation A.82), the following system matrices are obtained for the open loop system :

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -1 & 1 \\ 0 & 0 & -5 \end{bmatrix},\ b = \begin{bmatrix} 0 \\ 0 \\ 205 \end{bmatrix},\ c = [1\ 0\ \ 0],\ d = 0$$

Therefore, with the poles of the closed loop system given by : $p = [-5 \quad -4 + j5 \quad -4 - j5]$, the feedback matrix k is found by the following MATLAB command,

$$k = \text{acker}(A, b, p)$$
$$= [1\ 0.3366\ 0.0341]$$

The step response of the closed loop system shows an over shoot of 1.21% and settling time of $t_s = 0.771$ sec.

RE9.2 Figure 9.2 shows a schematic diagram of a communication satellite control system around one axis which is directed perpendicular to the page.

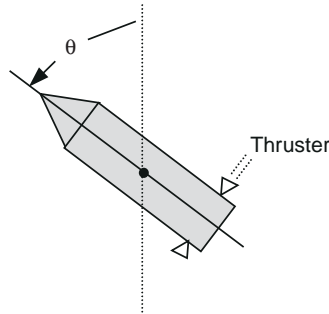



Fig. RE9.2

The differential equation is given by : $I\ddot{\theta} = T_c + T_d$

where I is the moment of inertia of the satellite around its center of mass, T_c is control torque, T_d is the disturbance torque, θ is the angle the satellite axis makes with an inertial reference line (assumed to have no acceleration). The above equation may be written as :

$$\ddot{\theta} = u + w_d$$

where $u = T_c/I$ and $w_d = T_d/I$

The transfer function of satellite attitude control system with $w_d = 0$ becomes:

$$G(s) = \frac{\theta(s)}{U(s)} = \frac{1}{s^2}$$

(*a*) Find the feedback gain matrix K that yields closed-loop roots with $\omega_n = 4$ rad/sec and δ = 0.65.

(*b*) Find the roots obtained by using LQR with

$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 20 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 50 & 0 \\ 0 & 5 \end{bmatrix}$$

and $R = 1$.

Solution: With the choice of states $\theta = x_1$, $\dot{\theta} = x_2$, the corresponding matrices in Equation (9.3) for the given plant transfer function G(s) are found as:

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, b = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, c = [1, 0], d = 0$$

(a) The desired characteristic equation with $\omega_n = 4$ and $\delta = 0.65$ is given by:

$$\alpha_c(s) = s^2 + 2\delta\omega_n s + \omega_n^2 = s^2 + 5.2s + 16 = 0.$$

Using the, the MATLAB script

```
A = [0 1 ; 0 0]
b = [0 ; 1]
Wn = 4
delta = 0.65
p = roots ([1  2*Wn*delta  Wn^2])
k = acker (A,b, p)
```

We get $p_1 = -1.6250 + 1.8998i$ and $p_2 = -1.6250 - 1.8998i$

The feedback control coefficients are found as : k = [6.2500 3.2500]

(b) Now using the scripts

```
Q = [1 0 ; 0 0],  [20 0; 0 1], and  [50 0 ; 0 5]
R = 1
k = lqr (A, b, Q, R)
p = eig(A – b * K)
[Wn, delta] = damp(p)
```

we get the respective feedback matrices k' = [1.0000 1.4142], [4.4721 3.1535], and [7.0711 4.3752]. which yields natural frequencies of ω_n = 1.0000, 2.1147, and 2.6591 rad/sec and damping of δ = 0.7071, 0.7456, and 0.8227 respectively.

RE9.3 For the transfer function $G(s) = \dfrac{1}{s^2}$

(a) find the feedback coefficients K to assign closed loop poles at $-1.5 \pm j\,2$ assuming all the states are available for measurement and feedback.

(b) Design an estimator for the satellite control system with estimator poles located at $-5 \pm j\,8$. Plot the estimator error response with initial values $e'(0)$ = [0 1].

Ans: (a) k' = [6.2500 3.0000]

RE9.4 For the open-loop system $G(s) = \dfrac{U(s)}{Y(s)} = \dfrac{1}{s^2 + 0.25\,s + 1}$

(a) Find the full state feedback that provides s-plane poles at ω_n = 2.5 rad/sec with δ = 0.6.

(b) Find $y(t)$ for the closed loop system with $y(0)'$ = [1 0] and the settling time to $y_1(0) = 0.2$.

Hints: Use the following script for the solution of RE 9.4

```
A=[0 1;-1 -0.25];b=[0; 1]; c=[1 0], d=0;
wn=2.5; delta=0.6;
p=roots([1 2*wn*delta wn^2]);
k=acker(A,b,p)
Af=A-b*k;
sys=ss(Af,b,c,d)
y0=[1;0];
t=0:0.01:5;
y=initial(sys,y0,t);
plot(t, y);
```

Ans: k=[5.2500 2.7500], t_s= 2.38

PROBLEMS

P9.1 Design an reduced order estimator for RE9.3 when x_1(t) is measured and $x_2(t)$ is estimated. Select the estimator pole at $s = -5$. Plot the estimator error response with $\dot{e}(0) = 1$.

Find the equivalent transfer function $D_{cs} = -\frac{U(s)}{Y(s)}$ for comparison with classical compensator from relation (9.49).

P9.2 If the system in RE9.3 is configured as in Fig. 9.8 to incorporate a reference input, find the matrix B_r and show that the step response does not affect the state estimator error.

P9.3 For a field controlled dc motor with transfer function given by : $G(s) = \frac{K}{s(s+5)(s+1)}$ find the feedback coefficients to place closed loop poles at $s = -2 \pm j\,6$. Also select K such that it meets the steady state requirements of $K_v > 220$.

P9.4 For the open-loop system $G(s) = \frac{2.5}{s^2}$

(*a*) Find the feedback coefficients to place the closed loop poles at $s = -2 \pm j3$ when both the states are available for measurement and feedback.

(*b*) Also design a full estimator when the estimator poles are located at $s = -6 \pm j6$.

(*c*) Obtain the representation of the complete system from (*a*) and (*b*).

P9.5 For the open-loop system $G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^2(s+2.5)}$

(*a*) Find the feedback coefficients such that the settling time, within 2% of the final value with a step input, is less than 1 sec. Assume that the damping ratio of the dominant roots is 0.625.

(*b*) Verify that t_s is satisfied by plotting the response to an initial value of y.

P9.6 For the system in P9.5 find the estimator equations and the value of the gain matrix G, for (*a*) full estimator with poles at $s = -1 \pm j0.8$, and (*b*) a reduced-order estimator ($\alpha_e(s) = s + 0.5 = 0$).

P9.7 For the open-loop system $G(s) = \frac{1}{s^2(s^2+16)}$

(*a*) find the feedback coefficients such that the closed loop poles are located at $s = -1 \pm j1.2$ and $s = -0.4 \pm j0.5$

(*b*) Design an estimator with poles located at $s = -2.5 \pm j2.5$, $s = -2.5 \pm j10$

(*c*) Also obtain the complete closed loop system representation incorporating (*a*) and (*b*).

P9.8 The dynamics of a magnetic levitation is described by the equations:

$$\ddot{x} = 900x + 15u$$

(*a*) Use pole placement design for this system such that the settling time is less than 0.4 sec and overshoot to an initial offset in x is less than 15% .

(*b*) Design a reduced-order estimator for $\dot{x}$ for this system such that the error-settling time will be less than 0.1 sec

(*c*) Plot step responses of x, $\dot{x}$ and u for an initial displacement $x(0)' = [1\ 0]$.

(*d*) Introduce a command reference as in Fig. 9.8. Compute the frequency response from r to system error $(r - x)$ and find the highest frequency for which the error amplitude is less than 15% of the command amplitude.

P9.9 For the double-mass spring system in Fig. A.51 in the Appendix A, if M = 1 kg, m = 0.1 Kg, β = 0.004 N-sec/m and k = 0.80 N/m, find the state feedback coefficients to place closed loop poles located at $s = -1 \pm j2;\ -4 \pm j6$. Also find the response $x_1(t)$ with initial value of $x_1(0) = 1$ m.

CHAPTER

10

Sampled Data Control System

10.0 WHY WE ARE INTERESTED IN SAMPLED DATA CONTROL SYSTEM ?

So far we have considered mostly continuous systems to introduce the basic principles of control engineering including their analysis and design in frequency and time domains. During the last six decades the conventional and modern control theory for continuous-time control system has revolutionized the industrial process and has enabled scientists and engineers to initiate exploration of space, which demands exacting system performance. Although, sampled data or discrete-time control theory has been evolving for the last five decades, its application to the development of practical systems has been slow due to inherent theoretical and physical implementation problems. However, the advent of microprocessors and embedded systems and their use as control elements have met satisfactorily the stringent performance specifications needed in space application.

In the literature, the terms *digital control, sampled data control and discrete data control systems* are used interchangeably. However, the digital data produced by digital computers may contain some sort of coding where as the sampled data are literally obtained from analog signals modulated by a train of pulses and hence are pulse-amplitude modulated signal. In a control system, both analog and digital signals will be present because plants are mostly analog in nature. The term *discrete data systems* will be used to imply both digital data or sampled data systems. In general, systems which utilize digital computers as control elements will be referred to as digital control or discrete data systems.

Even though the use of analog controllers are quite old and it has served the cause of industry quite satisfactorily, its digital counterpart has the following advantages.

10.1 ADVANTAGE OF DIGITAL CONTROL

(*a*) **It gives better resolution:** The resolution of representing number by digital technique depends on the number of bits, which can be increased theoretically without limit (Example: RPM measurement using Tachometer and digital methods, LDR and Light source).

(*b*) **Coding of digital signal and digital communication:** The entry of digital control to the industrial scenario was chiefly through the big process industries like steel mills, chemical, fertilizer and cement industries because of the cost advantage. In a modern steel mill, which might spread over an area of about 10 Km radius, typical computing power needed are: two mainframe computers (one for the head office) several tens of minicomputers, several hundreds PC's and several thousand micro-controllers. These computers are configured in a hierarchical fashion and lots of data communications are involved among them. When using digital control, the communication of data can be undertaken in digital mode. So, it is advantageous to use

digital control, since digital communication is more reliable due to its improved noise immunity compared to its analog counterpart.

(*c*) **Time sharing:** One communication channel may be used to transmit discrete control signal for more than one control systems. A well known application of sampled-data process is the time sharing of a communication channel for transmitting measured data from space-vehicle. Instead of temperature at five different places in the vehicle and pressure at three different places being telemetered back on eight different channels, each signal uses the same channel for one second, say out of eight.

(*d*) **Control system with inherent sampling:** Sampling is inherent in some applications (*e.g.,* radar scanning), and it is necessary in other applications (*e.g.,* to enter data into digital computers), it is desirable for still other applications (*e.g.,* time sharing). In a guidance control system that utilizes radar scanning in which a given sector or region is scanned once every revolution.

Thus in any direction, the signal is sampled at a rate equal to the scan rate of the radar.

(*e*) **Digital computer implementation of sophisticated control law:** Dedicated mini computers and microprocessors can be used to implement sophisticated algorithms such as high performance state variable manipulation, Kalman Filtering and Adaptive control and the like. The use of digital computer as a controller has offered the designer tremendous amount of flexibility and versatility in the design approach.

(*f*) **System design:** For some control system application, better system performance may be achieved by a sampled data control system design. For example, one method of neutralizing the effect of transportation lag which degrades the relative stability of the system, is to insert a pure differentiation unit in the forward loop of the control system. For a continuous system the differentiator, not only may enhance the existing noise but may also be a source of additional noise. In a sampled data system, the differentiating action can be implemented by a digital computer without increasing system noise problem.

The foregoing advantages of a digital controller are, however, accompanied by some of the disadvantages mentioned below:

10.2 DISADVANTAGES

(*a*) **System design:** The mathematical analysis of a digital controller is sometimes more complex and more tedious as compared to continuous data control systems.

(*b*) **System stability:** System stability is degraded by converting a continuous data system to digital system. Even a stable second order system can be unstable when converted to discrete version.

(*c*) **Loss of signal information:** Even if we employ digital computer as a controller, most of the processes are analog in nature, which needs analog actuators. So, the signal output from the digital controller is to be converted to analog form before applying to the actuators. However, the reconstruction of digital to analog signal using hold circuits is only an approximation of the actual signal. So, some signal information is lost in the process.

(*d*) **Controller dynamic update:** Analog to Digital (A/D) and Digital to Analog (D/A) converters along with digital computer introduce some time delay. Hence the performance objectives may be difficult to achieve since the theory assumes no delay.

(*e*) **Software errors:** The complexity of the controller is in the software implemented algorithms, which may contain errors.

The extensive use of digital computers and micro controllers make it imperative to present a mathematical tool that can handle analysis and design of sampled data systems. In the following section, we present one such tool—the Z-transform together with some of its properties in the form of theorems.

10.3 REPRESENTATION OF SAMPLED PROCESS

The sampling process is schematically represented in Fig. 10.1, where $f(t)$ is an analog signal whose amplitude is sampled at *fixed intervals of period* T by a sampler represented by the switch to produce the sampled signal $f^*(t)$. The value of $f^*(t)$ is the amplitude of the analog signal measured during the time the switch remain closed [vide Fig. 10.1 (*b*)].

When the pulse duration is much shorter than the fastest time constant of $f(t)$, the output of the sampler may be approximated by the train of impulses $f^*(t)$. The term $f^*(t)$ is read as *f* star of *t*. The strength (area) of each impulse is equal to the value of the input analog signal at the time $t = n$T of the particular impulse [Fig. 10.1 (*c*)].

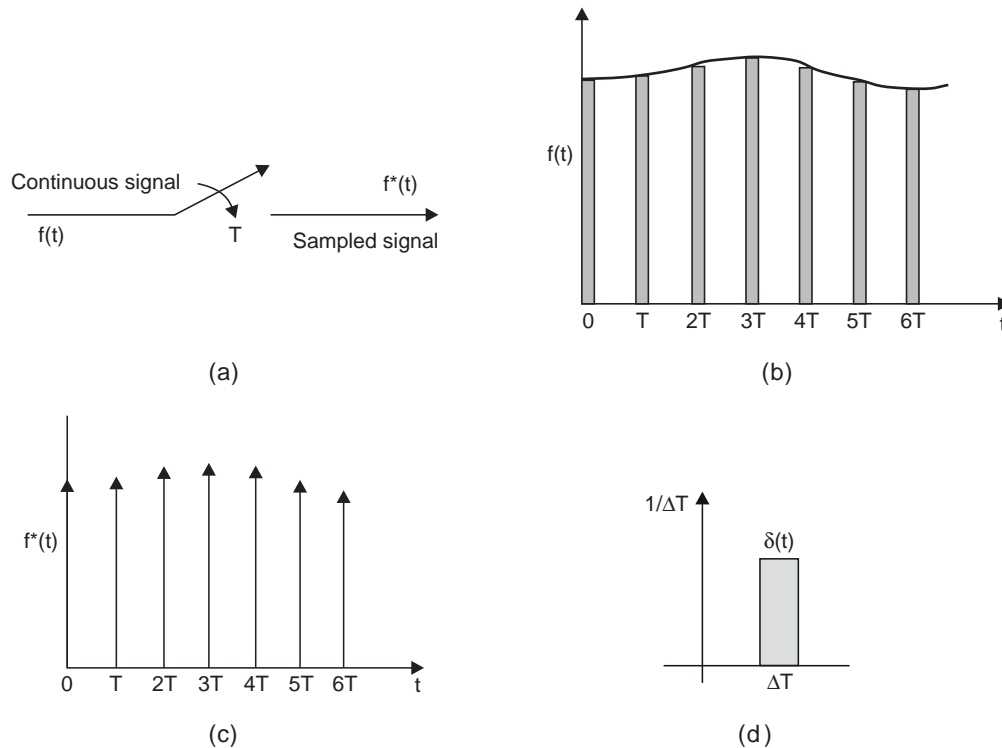



Fig. 10.1 (a) Sampler switch (b) continuous input $f(t)$ and output of switch (shaded pulses) (c) impulse approximation of switch output $f^*(t)$ (d) unit Impulse

Thus the area of the n^{th} impulse which occurs at time $t = n$T is $f(n\text{T})$. The equation for the entire train of impulses may be represented as

$$f^*(t) = f^*(0)\,\delta(t) + f^*(\text{T})\,\delta(t - \text{T}) + f^*(2\text{T})\,\delta(t - 2\text{T}) + \ldots \qquad (10.1)$$

$$= \sum_{n=0}^{\infty} f(n\text{T})\,\delta(t - n\text{T})$$

where $\delta(t)$ is unit impulse at $t = 0$ and $\delta(t - n\text{T})$ is a unit impulse at $t = n$T.

The Laplace transform of the sampled signal is

$$F^*(s) = £\,[f^*(t)] = f(0) + f(T)\,e^{-Ts} + f[(2T)]e^{-2Ts} + \ldots\ldots$$

$$= \sum_{n=0}^{\infty} f(nT)\,e^{-nTs} \tag{10.2}$$

Here £ is the Laplace operator. The Laplace transform $F^*(s)$ contains terms of the form e^{-sT} and is not a rational function of s. The complex function $F^*(s)$ can be converted to convenient form by transforming to another complex domain z by making a substitution.

10.4 THE Z-TRANSFORM

The simple substitution of $z = e^{sT}$ or $s = \frac{1}{T}\ln z$

converts the Laplace transform to the z-transform. Making this substitution in Equation (10.2) gives

$$Z[f^*(t)] = F(z) = f(0) + \frac{f(T)}{z} + \frac{f(2T)}{z^2} + \ldots$$

or

$$F(z) = \sum_{n=0}^{\infty} f(nT)\,z^{-n} \tag{10.3}$$

where Z is the z-transform operator and $F(z)$ designates the z-transform of $f^*(t)$. Because only values of the signal at the sampling instants are considered, the z-transform of $f(t)$ is the same as that of $f^*(t)$.

So we note that $$F(z) = \begin{cases} \text{Laplace transform of } f^*(t)\,|_{s=\ln(z)/T} \\ F^*(s)\,|_{s=\ln(z)/T} \end{cases}$$

So we can compute the z-transform directly from sampled function in time domain or $f^*(t)$ or its Laplace transform $F^*(s)$. We formally define z-transform as follows:

If a signal has discrete values $f(0), f(T), f(2T) \ldots, f(kT)$, the z-transform for the signal is defined as

$$F(z) = \sum_{n=-\infty}^{\infty} f(nT)\,z^{-n},\; r < |\,z\,| < R \tag{10.4}$$

where r and R are the lower and upper limits of the complex variable z for which the series converges.

If the signal is such that $f(n) = 0$ for $n < 0$ (*i.e.,* does not exist for negative time), then $R \to \infty$ and the region of convergence is the whole z-plane outside a circle of finite radius r and $F(z)$ will be given by (10.3) . If $f(n) \neq 0$ for $n < 0$, then R is finite.

For instance, if $F(n) = -1$ for $n < 0$ and $f(n) = 0$ for $n \geq 0$, it can be shown that $F(z)$ will converge to $z/(z-1)$ for $|z| < 1$.

Some examples are now considered to illustrate the computation of z-transform of some typical functions.

Example 10.1 Determine the z-transform for a unit step function.

For this function $f(n\text{T}) = u(n\text{T}) = 1$ for $n = 0, 1, 2, \ldots\ldots.$; and $\text{F}(s) = \dfrac{1}{s}$. Thus application of Equation (10.3) gives

$$z \text{ transform of } [u^*(t)] = 1 + \frac{1}{z} + \frac{1}{z^2} + \ldots.$$

$$= \left(1 - \frac{1}{z}\right)^{-1} = \left(\frac{z-1}{z}\right)^{-1} = \frac{z}{z-1} \tag{10.5}$$

This series is convergent for $1/z < 1$.

Example 10.2 Let us find the z-transform of the exponential function e^{-at}. For this function $f(n\text{T}) = e^{-ant}$ and $\text{F}(s) = \dfrac{1}{s+a}$. Therefore,

$$z\text{-transform of } (e^{-at}) = 1 + \frac{e^{-at}}{z} + \left(\frac{e^{-a\text{T}}}{z}\right)^2 + \ldots = \left(1 - \frac{e^{-a\text{T}}}{z}\right)^{-1} = \left(\frac{z - e^{-a\text{T}}}{z}\right)^{-1} = \left(\frac{z}{z - e^{-a\text{T}}}\right) \tag{10.6}$$

Example 10.3 Find the z-transform for the function whose Laplace transform is given by

$$\text{F}(s) = \frac{1}{s(s+1)} = \frac{1}{s} - \frac{1}{(s+1)}$$

Using the results of Examples 10.1 and 10.2, we can write,

$$\text{F}(z) = \frac{z}{z-1} - \frac{z}{z - e^{-T}} = \frac{z(1 - e^{-\text{T}})}{(z-1)(z - e^{-\text{T}})} \tag{10.7}$$

10.4.1 The Residue Method

The residue method is a powerful technique for obtaining z-transforms of the function $f^*(t)$ when its Laplace transform is available. The z-transform of $f^*(t)$ may be expressed in the form

$$\text{F}(z) = z\text{-transform of } [f^*(\text{T})] = \Sigma \text{ residues of F}(s)\, z/(z - e^{s\text{T}}) \text{ at poles of F}(s) \tag{10.8}$$

When the denominator of F(s) contains a linear factor of the form $(s - r)$ such that F(s) has a first-order pole at $s = r$, the corresponding residue R is found as

$$\text{R} = \lim_{s \to r} (s - r) \left[\text{F}(s) \frac{z}{z - e^{s\text{T}}}\right] \tag{10.9}$$

When F(s) contains a repeated pole of order q, the residue is computed as

$$\text{R} = \frac{1}{(q-1)!} \lim_{s \to r} \frac{d^{q-1}}{ds^{q-1}} \left[(s-r)^q \,\text{F}(s) \frac{z}{z - e^{s\text{T}}}\right] \tag{10.10}$$

Example 10.4 Determine the z-transform of a unit step function. In this case, F(s) = $1/s$, which has one pole at $s = 0$; The corresponding residue is

$$\text{R} = \lim_{s \to 0} s \left(\frac{1}{s} \frac{z}{z - e^{s\text{T}}}\right) = \frac{z}{z-1}\text{ ; Therefore, F}(z) = \frac{z}{z-1}$$

Example 10.5 Consider the function F(s) given by

$$F(s) = \frac{1}{s+a}\ ;\ \text{The pole is at } -a.$$

Hence, $$R = \lim_{s \to -a} (s+a)\left[\frac{1}{s+a} \cdot \frac{z}{z-e^{sT}}\right] = \frac{z}{z-e^{-aT}} \text{ and } F(z) = \frac{z}{z-e^{-aT}}.$$

Example 10.6 Consider the function $F(s) = \dfrac{1}{s(s+1)}$

The poles are at $s_1 = 0$ and $s_2 = -1$. The residue at $s_1 = 0$ is given by

$$R_1 = \lim_{s \to 0} s\left[\frac{1}{s(s+1)} \frac{z}{z-e^{sT}}\right] = \frac{z}{z-1}$$

The residue at $s_2 = -1$ is found as $R_2 = \lim_{s \to -1} (s+1)\left[\dfrac{1}{s(s+1)} \dfrac{z}{z-e^{sT}}\right] = \dfrac{-z}{z-e^{-T}}$

Therefore, $$F(z) = R_1 + R_2 = F(z) = \frac{z}{z-1} - \frac{z}{z-e^{-T}}$$

Example 10.7 Consider a function with complex conjugate poles given by:

$$F(s) = \frac{s}{s^2+\omega^2} = \frac{s}{(s-j\omega)(s+j\omega)}, \text{ corresponding to } f(t) = \cos \omega t$$

Thus $$R_1 = \lim_{s_1 \to j\omega} (s-j\omega)\left[\frac{s}{(s-j\omega)(s+j\omega)} \frac{z}{z-e^{sT}}\right] = \frac{j\omega}{2j\omega} \frac{z}{z-e^{j\omega T}} = \frac{1}{2} \cdot \frac{z}{z-e^{j\omega T}}$$

and $$R_2 = \lim_{s_2 \to -j\omega} \left[\frac{s}{s-j\omega} \frac{z}{z-e^{sT}}\right] = \frac{-j\omega}{-2j\omega} \frac{z}{z-e^{-j\omega T}} = \frac{1}{2} \cdot \frac{z}{z-e^{-j\omega T}}$$

$$\therefore \quad F(z) = \frac{1}{2}\left[\frac{z}{z-e^{j\omega T}} + \frac{z}{z-e^{-j\omega T}}\right] = \frac{z^2 - z\dfrac{(e^{-j\omega T}+e^{j\omega T})}{2}}{z^2 - 2z\dfrac{(e^{-j\omega T}+e^{j\omega T})}{2} + 1} = \frac{z^2 - z\cos \omega T}{z^2 - 2z\cos \omega T + 1}.$$

Example 10.8 Let us now consider a function with multiple poles.

Let, $f(t) = t \quad \therefore \quad F(s) = 1/s^2$, So, F(s) has a second order pole at $s = 0$.

Using relation (10.10) with $q = 2$, we find $F(z) = \dfrac{Tz}{(z-1)^2}$

Similarly for $F(s) = \dfrac{1}{s^3}$ corresponding to $f(t) = \dfrac{t^2}{2}$, the z-transform is found to be

$$F(z) = \frac{T^2}{2} \frac{z(z+1)}{(z-1)^3}$$

Some important theorems are presented, without proof, which are useful for finding z-transforms of some functions.

10.4.2 Some Useful Theorems

(*i*) Multiplication by e^{-at}

$$z\text{-transform of } [e^{-at} \; f(t)] = F(z\, e^{-at}) \tag{10.11}$$

(*ii*) Multiplication by t.

$$z\text{-transform of } t\, f(t) = -\, z\mathrm{T}\; d/dz\; [F(z)] \tag{10.12}$$

(*iii*) Multiplication by $a^{t/\mathrm{T}}$

$$z\text{-transform of } [a^{t/\mathrm{T}} f(t)] = F(z/a) \tag{10.13}$$

(*iv*) Partial differentiation. This theorem states that

$$z\text{-transform of } [\partial/\partial a[f(t, a)]] = \partial/\partial a\; [F(z, a)] \tag{10.14}$$

(*v*) Multiplication by a^k

$$z\text{-transform of } [a^k f(k)] = F(z/a) \tag{10.15}$$

(*vi*) Initial Value Theorem $f(0) = \lim\limits_{z \to \infty} F(z)$ (10.16)

(*vii*) Final Value theorem $f(\infty) = \lim\limits_{z \to 1} (1 - z^{-1})F(z)$ (10.17)

(*viii*) Real translation (shifting theorem)

If $f(t)$ has the z-transform $F(z)$ then (*a*) z-transform of $f(t - n\mathrm{T}) = z^{-n}\, F(z)$ (10.18)

and (*b*) z-transform of $f(t + nt) = z^n\, F(z) - \sum\limits_{i=1}^{N} z^i\, f[(n - i)\,\mathrm{T}]$ (10.19)

where n is a positive integer

Initial Value Theorem

With the help of initial value theorem we can find the area of the first impulse, $f(0)$, of the sampled function $f^*(t)$ without performing the inverse operation on the function $F(z)$.

If $f(t)$ has the z-transform $F(z)$ and $\lim\limits_{z \to \infty} F(z)$ exists, then the initial value $f(0)$ of $f(t)$ will be given by

$$f(0) = \lim_{z \to \infty} F(z)$$

To prove this, we note that $\quad F(z) = \sum\limits_{n=0}^{\infty} f(n\mathrm{T})z^{-n} = f(0) + f(\mathrm{T})z^{-1} + f(2\mathrm{T})z^{-2} + \ldots$

Now with $z \to \infty$, we find $\quad \lim\limits_{z \to \infty} F(z) = f(0)$

Final Value Theorem

With the help of final value theorem we can compute the value of $f^*(t)$ as t approaches infinity without performing the inverse operation on $F(z)$. It will be found to be very useful in assessing the steady state performance of a control system.

The area of the impulse $f(n\mathrm{T})$ as n becomes infinite is

$$f(\infty) = \lim_{z \to 1} \{(z - 1)/z\}F(z)$$

$$= \lim_{z \to 1} (1 - z^{-1})F(z)$$

To prove the final value theorem, we write the pulse sequence $f^*(t)$ truncated at the Nth sample where N is some large number as follows:

$$F_N(z) = \sum_{n=0}^{N} f(nT)\, z^{-n}$$

The truncated sequence is shown in Fig. 10.2 (*a*). If the truncated function is now delayed by one sample time T, then $f^*_N(t - T)$ is formed and the sequence is plotted in Fig. 10.2 (*b*).

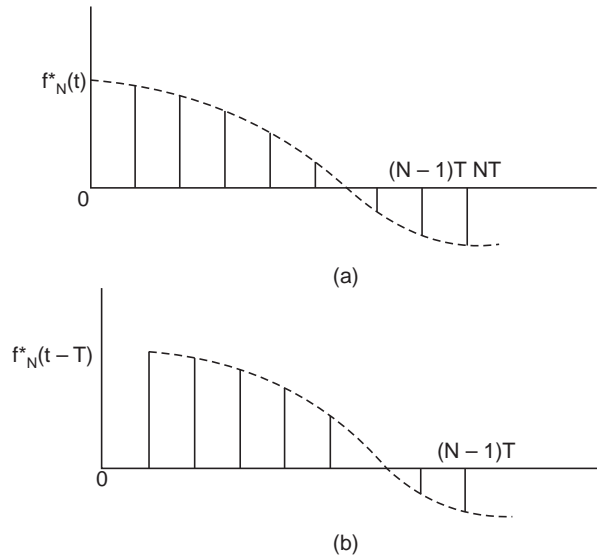



Fig 10.2 (a)The function $F_N^*(t)$ is a sampled function truncated at t = NT
(b) $F_N^*(t)$ shifted by one sampling period

It is evident that the *z*-transform of the delayed function is the same as that of the truncated function except that an additional delay z^{-1} is included as factor.

Thus
$$F'_N(z) = z^{-1}F_N(z) = \sum_{n=0}^{N-1} f(nT)\, z^{-n} \,.\, z^{-1} \qquad (10.20)$$

It is noted that the sequence terminates at N – 1 since the absolute time t of truncation remains fixed at NT.

Therefore

$$F_N(z) = f(0) + f(T)\, z^{-1} + f(2T)\, z^{-2} + \ldots\ldots + f[(N-1)T]\, z^{-(N-1)} + f(N)\, z^{-N}$$
$$F_N'(z) = f(0)\, z^{-1} + f(T)\, z^{-2} + f(2T)\, z^{-3} + \ldots\ldots + f[(N-1)T]\, z^{-N}$$

....

$$F_N(z) - F_N'(z) = f(0)\,(1 - z^{-1}) + f(T)z^{-1}\,(1 - z^{-1}) + f(2T)\, z^{-2}\,(1 - z^{-1}) + f[(N-1)T]z^{-(N-1)}[1 - z^{-1}] + f(NT)\, z^{-N}$$

Therefore,
$$f(NT) = \lim_{z \to 1} \{F_N(z) - F_N'(z)\} = \lim_{z \to 1} \left\{ \sum_{n=0}^{N} f(nT)\, z^{-n} - z^{-1} \sum_{n=0}^{N-1} f(nT) z^{-n} \right\}$$

Chapter 10

Now as N is allowed to increase without limit and approaches infinity, it is seen that the two summations above each coverage to $F(z)$ because, in the limit N and N – 1 coverage toward the same value. The final value theorem can be stated as follows in consequence of the defining relation (10.3).

Hence, $$f(\infty) = \lim_{z \to 1} (1 - z^{-1}) F(z)$$

where $f(\infty)$ is the final value of the sample of the sequence $f^*(t)$ whose z-transform is $F(z)$. This theorem is of paramount importance to the designer of sampled data control systems for evaluating steady state performance.

Real Translation

The solid curve shown in Fig. 10.3 (*a*) is the continuous function $f(t)$. The value at time $t = 0$ is $f(0)$. The value at t is $f(t)$ and so on. When the function is shifted to the right (delayed) by a time nT, as shown in Fig. 10.3 (*b*), the function $f(t - nT)$ results. The value of $f(t - nT)$ at time nT is $f(0)$, the value at $t = (n + 1)T$ is $f(T)$ etc.

The z-transformation of $f(t - nT) = z^{-n} F(z)$

Similarly, z-transform of $[F(t + nT)] = z^n F(z) - \sum_{i=1}^{n} z^i [f(n - i)T]$

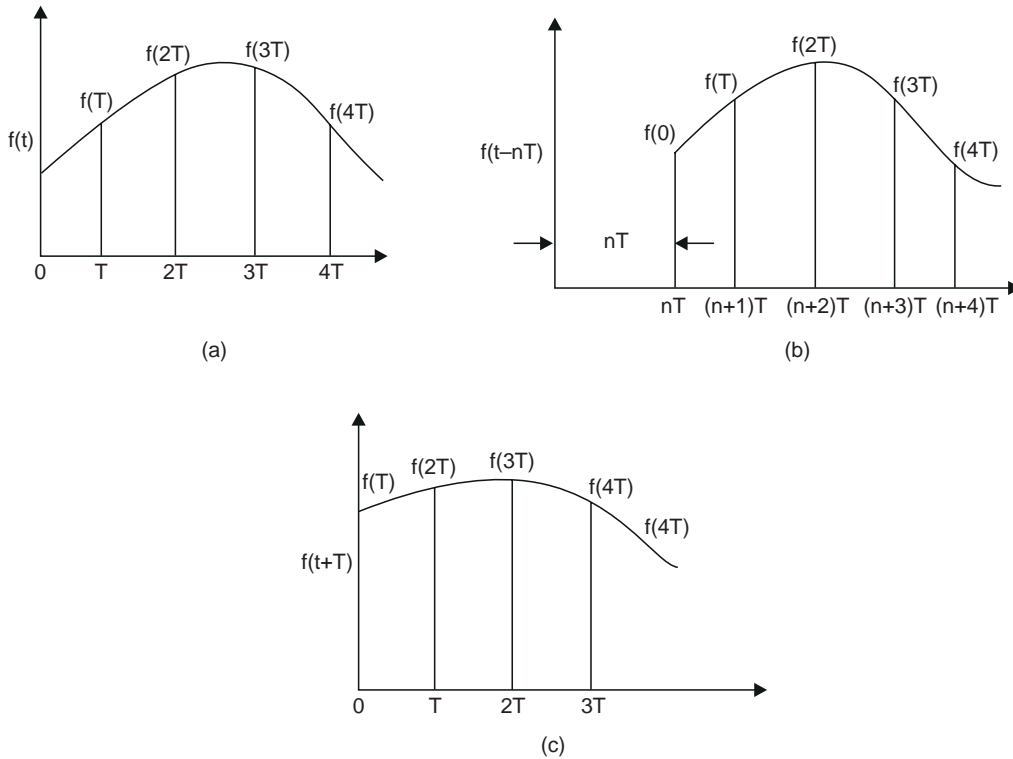



Fig. 10.3 Translation of a continuous function

Proof of shifting theorem

We shall prove that if $f(\mathrm{T})$ has the z–transform $\mathrm{F}(z)$ then

(*i*) z-transform of $f(t - n\mathrm{T}) = z^{-n}\,\mathrm{F}(z)$ and

$$(ii)\ z\text{-transform of } f(t + n\mathrm{T}) = z^{n}\,\mathrm{F}(z) - \sum_{i=1}^{n} z^{i} f[(n-i)\mathrm{T}]$$

where n is a positive integer.

Proof: By definition

$$z \text{ of } f(t - n\mathrm{T}) = \sum_{k=0}^{\infty} f(k\mathrm{T} - n\mathrm{T})z^{-k} = z^{-n} \sum_{k=0}^{\infty} f(k\mathrm{T} - n\mathrm{T})z^{-(k-n)}$$

Substituting $m = k - n$, so that $m = -n$ when $k = 0$, and recalling that $f(k) = 0$ for $k < 0$, the above expression becomes:

$$z^{-n} \sum_{m=-n}^{\infty} f(m\mathrm{T})\, z^{-m} = z^{-n} \sum_{m=0}^{\infty} f(m\mathrm{T})z^{-m} = z^{-n}\,\mathrm{F}(z),$$

which proves the first part of the theorem.

$$\text{Now } z\text{-transform of } [f(t + n\mathrm{T})] = \sum_{k=0}^{\infty} f(k\mathrm{T} + n\mathrm{T})\, z^{-k}$$

$$= z^{n} \sum_{k=0}^{\infty} f(k\mathrm{T} + n\mathrm{T})z^{-(k+n)} = z^{n} \sum_{m=n}^{\infty} f(m\mathrm{T})z^{-m}$$

$$= z^{n} \left[\sum_{m=0}^{\infty} f(m\mathrm{T})z^{-m}\right] - \sum_{m=0}^{n-1} f(m\mathrm{T})z^{n-m}$$

$$= z^{n}\,\mathrm{F}(z) - \sum_{m=0}^{n-1} f(m\mathrm{T})\, z^{n-m} = z^{n}\, F(z) - \sum_{i=n}^{1} f[(n-i)\mathrm{T}]z^{i}$$

$$= z^{n}\,\mathrm{F}(z) - \sum_{i=1}^{n} z^{i} f[(n-i)\mathrm{T}],$$

which proves the second part of the theorem.

Some applications of the above theorems are now considered.

We know that the z-transform of a unity step for which $f^*(k\mathrm{T}) = 1$, is given by $\mathrm{F}(z) = \dfrac{z}{z-1}$; hence z-transform for the function $f^*(k\mathrm{T}) = a^{k}$, can be computed by theorem (v) [in Section 10.4.2] as

$$z\text{-transform of } f(1\,.\,a^{k}) = \mathrm{F}\left[\frac{z}{a}\right] = \frac{\dfrac{z}{a}}{\dfrac{z}{a} - 1} = \frac{z}{z-a}$$

Example 10.9 Multiplication by t

For a unit step function, $f(k\text{T}) = 1$, and $\text{F}(z) = z/(z-1)$. The z-transform of $t\,f(t) = t\,.\,1$ can be found by the theorem (*ii*) [in Section 10.4.2] as

The z-transform of $t\,f(t) = -z\text{T}\,\dfrac{d\text{F}(z)}{dz} = z\,\text{T}/(z-1)^2$ where, $\text{F}(z)$ is the z-transform of $f(t)$

10.5 INVERSE Z-TRANSFORMS

10.5.1 Partial Fraction Method

The function F(z) is initially expanded in partial fraction expansion then each term is identified with standard entries in z-transform table (Table 10.1) for writing the inverse z transform.

As illustrative example, let

$$\text{F}(z) = \frac{(1-e^{-\text{T}})z}{(z-1)(z-e^{-\text{T}})} = \left[\frac{k_1}{z-1} + \frac{k_2}{z-e^{-\text{T}}}\right] = \frac{z}{z-1} - \frac{z}{z-e^{-\text{T}}}$$

Comparing with standard terms from Z-transform, Table 10.1 (row 3 and 7), we find

$$f(t) = u_s(t) - e^{-t}$$

Therefore,
$$f^*(t) = f(n\text{T}) = \sum_{n=0}^{\infty} (1 - e^{-n\text{T}})\,\delta(t - n\text{T})$$

10.5.2 Residue Method

The inverse z-transform $f(n\text{T})$ from F(z) can be found as: $f(n\text{T}) = \Sigma$ residues of $\text{F}(z)\,z^{n-1}$ at poles of $\text{F}(z)\,z^{n-1}$ The residue to a repeated pole at $z = r$ of order q is given by:

$$\text{R} = \frac{1}{(q-1)!} \lim_{z \to r} \frac{d^{q-1}}{d_z^{q-1}} [(z-r)^q\, \text{F}[z]\, z^{n-1}]$$

Example 10.10 Let us find the inverse z-transform of F(z) given by:

$$\text{F}(z) = \frac{(1-e^{-\text{T}})z}{(z-1)(z-e^{-\text{T}})}$$

We note that F(z) has two poles located at $z = 1$ and $z = e^{-\text{T}}$

The residue, R_1 at the first pole is computed as $\text{R}_1 = \lim\limits_{z \to 1}\left[\left(\dfrac{1-e^{-\text{T}}}{z-e^{-\text{T}}}\right)z^n\right] = 1$

Similarly the second residue is found as $\text{R}_2 = \lim\limits_{z \to e^{-\text{T}}}\left[\left(\dfrac{1-e^{-\text{T}}}{z-1}\right)z^n\right] = -e^{-n\text{T}}$

Therefore, $\text{R}_1 + \text{R}_2 = u_s(n\text{T}) - e^{-n\text{T}} = f(n\text{T})$

Example 10.11 Consider a function F(z) with repeated poles as given below:

$$\text{F}(z) = \frac{\text{T}_z}{(z-1)^2}, \text{ F}(z) \text{ has double pole located at } -1$$

Therefore, the residue is found as $R = \lim_{z \to 1} \frac{d}{dz} (T z^n) = nT z^{n-1} \big|_{z=1} = nT$

So, the discrete domain function is $f(nT) = nT$ and the corresponding continuous domain function is $f(t) = t$.

10.6 BLOCK DIAGRAM ALGEBRA FOR DISCRETE DATA SYSTEM

In writing the transfer function for feedback control system with sampling switches, one encounters some terms which are sampled (starred) and some which are not. Thus it is necessary to develop some mathematical techniques for handling such mixed terms. We shall imagine placing a fictitious sampler at the output of any analog block (vide Fig. 10.4) to produce starred signal so as to ensure that signals at all points in the system are in sampled form. The sampling rate is assumed to be the same throughout the system and sampling is done at the same instant.

In Fig. 10.4(a) is shown a sampling switch followed by a linear element whose transfer function is $G(s)$, the transformed equation for the output $Y(s)$ is

$$Y(s) = F^*(s)\, G(s) \tag{10.21}$$

For $0 < t < T$, the response $y(t)$ is that due to the first impulse at $t = 0$ of area $f(0)$.

Thus, for this interval

$$y(t) = \text{Laplace inverse of } [f(0)\, G(s)] = f(0)\, £^{-1}\, [G(s)] = f(0)\, g(t) \tag{10.22}$$

where $g(t)$ = Laplace inverse of $[G(s)]$ is the response of the linear element to a unit impulse which occurs at $t = 0$.

For $T < t < 2T$, the response $y(t)$ is that due to the impulse at $t = 0$ plus that at $t = T$. For this interval,

$$F^*(s) = f(0) + f(T)\, e^{-Ts}$$

Thus, $$Y^*(s) = [f(0) + f(T)\, e^{-Ts}]\, G(s)$$

Taking Laplace inverse we get, $$y(t) = f(0)\, g(t) + f(T)\, g(t - T) \tag{10.23}$$

where $g(t - T) = £^{-1}\, [G(s)e^{-Ts}]$ is the response of the linear element to a unit impulse which occurs at time $t = T$.

The response $y(t)$ for the interval $2T < t < 3T$ is

$$y(t) = f(0)\, g(t) + f(T)\, g(t - T) + f(2T)\, g(t - 2T)$$

In general, $$y(t) = \sum_{n=0}^{\infty} f(nT) g(t - nT)\,; \quad y(kT) = \sum_{n=0}^{\infty} f(nT)\, g\,(kT - nT) \tag{10.24}$$

When n is such that $nT > t$, then $g(t - nT)$ is zero . That is, the impulse response is zero for negative time. Taking the limit as t approaches 0 in Equations (10.22),(10.23) and (10.24), we get the value of $y(nT)$ at the sampling instants:

$$\begin{aligned}
y(0) &= f(0)\, g(0)\\
y(T) &= f(0)\, g(T) + f(T)\, g(0)\\
y(2T) &= f(0)\, g(2T) + f(T)\, g(T) + f(2T)\, g(0)\\
&\vdots\\
y(nT) &= f(0)\, g(nT) + f(T)\, g[(n - 1)\, T] + \ldots
\end{aligned} \tag{10.25}$$

Replacing f by Y in Equation (10.2) gives

$$Y^*(s) = y(0) + y(T)\, e^{-Ts} + y(2T)\, e^{-2Ts} + \ldots$$

Substitution of the values from Equation (10.25) into the expression for $Y^*(s)$ gives

$$\begin{aligned} Y^*(s) = f(0)\, &[g(0) + g(T)\, e^{-Ts} + g(2T)\, e^{-2Ts} + \ldots.] \\ &+ f(T)\, e^{-Ts}\, [g(0) + g(T)\, e^{-Ts} + g(2T) e^{-2Ts} + \ldots] \\ &+ f(2T) e^{-2Ts}\, [g(0) + g(T)\, e^{-Ts} + g(2T)\, e^{-2Ts} + ..] \\ &+ f(nT)\, e^{-nTs}\, [g(0) + g(T)\, e^{-Ts} + g(2T)\, e^{-2Ts} + ..] \\ = [f(0) + f(T)\, e^{-Ts} &+ f(2T)\, e^{-2Ts} \ldots + f(nT)\, e^{-nTs}\,][g(0) + g(T) e^{-Ts} + g(2T) e^{-2Ts} + \ldots] \\ = F^*(s) G^*(s) & \qquad (10.26) \end{aligned}$$

Thus $Y^*(s) = F^*(s)\, G^*(s)$ (10.27)

The term $G^*(s)$ is called the pulse-transfer function of the system in Fig. 10.4 (*a*).

Comparison of Equations (10.21) and (10.27) reveal a basic mathematical relationship for starring qualities. That is, starring both sides of Equation (10.21) gives

$$\begin{aligned} [Y(s)]^* &= Y^*(s) \\ [F^*(s) G(s)]^* &= F^*(s) [G(s)]^* \\ &= F^*(s) G^*(s) \end{aligned}$$

Letting $z = e^{sT}$ in Equation (10.27) yields the z-transform relationship

$$Y(z) = F(z) G(z) \qquad (10.28)$$

For the sampler configurations of Fig. 10.4(*b*), the Laplace relationship is

$$Y(s) = F^*(s) G_1(s) G_2(s)$$

Starring gives

$$Y^*(s) = F^*(s) [G_1(s)\, G_2(s)]^* = F^*\,(s)\, G_1\, G_2{}^*(s),\ \text{where } G_1\, G_2{}^*\,(s) = [G_1(s)\, G_2(s)]^*$$

The corresponding z-transform is

$$Y(z) = F(z)\, G_1\, G_2(z) \qquad (10.29)$$

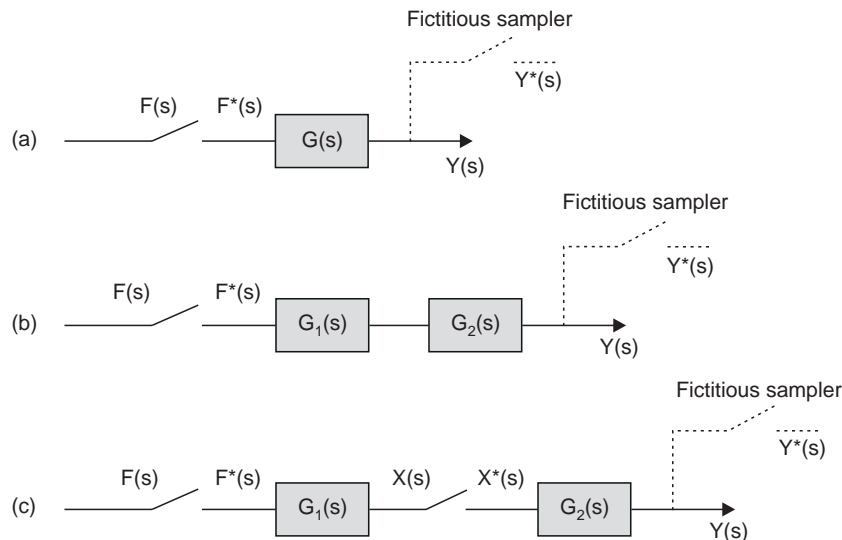



Fig. 10.4 Block diagrams with mixed signals

Example 10.12 Let us find the z-transform of the output signal in Fig. 10.4(b), when $G_1(s) = 1/s$ and $G_2(s) = 1/(s + 1)$. The product $G_1(s)G_2(s) = 1/s(s + 1)$

The Z-transform of this function is given by $G_1G_2(z) = z(1 - e^{-T})/[(z - 1)(z - e^{-T})]$

For the sampler configuration in Fig. 10.4(c), the Laplace relationship are:

$$X(s) = F^*(s)G_1(s) \text{ and } Y(s) = X^*(s)G_2(s)$$

Starring the first equation and then substituting this result for $X^*(s)$ into the second equation gives

$$Y(s) = [F^*(s)G_1(s)]^*\ G_2^*(s) = F^*(s)\ G_1^*(s)\ G_2^*(s)$$

The corresponding z-transform of the output signal is

$$Y(z) = F(z)G_1(z)G_2(z) \qquad (10.30)$$

Example 10.13 Determine the z-transform for Fig. 10.4(c), when $G_1(s) = 1/s$, $G_2(s) = 1/(s + 1)$; From the previous example it is known that

$$G_1(z) = z/(z - 1) \text{ and } G_2(z) = z/(z - e^{-T})$$

Thus $\qquad G_1(z)G_2(z) = z^2/(z - 1)(z - e^{-T})$

From the preceding two examples, it is to be noted that

$$G_1G_2(z) \neq G_1(z)G_2(z) \qquad (10.31)$$

Two sampled data feedback control systems are shown in Fig. 10.5. The general procedure for determining the transformed equation for a sampled-data system is:

(a) In addition to the actual system input R, regard all switch outputs (starred quantities) as inputs.

(b) In addition to the system output y, regard all switch inputs as outputs.

(c) Write equations for each output in terms of its inputs.

(d) Star quantities as necessary in order to determine $y(z)$.

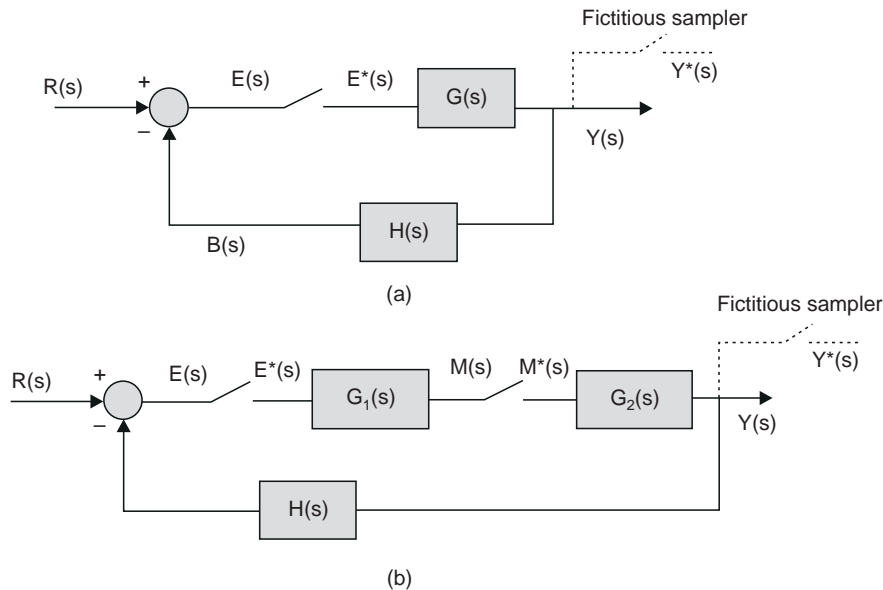



Fig. 10.5 Hybrid system

Application of this method to the system shown in Fig. 10.5 (*a*) gives

$$Y(s) = E^*(s)G(s)$$
$$E(s) = R(s) - E^*(s)G(s)H(s)$$

Starring gives

$$Y^*(s) = E^*(s)G^*(s)$$
$$E^*(s) = R^*(s) - E^*(s)GH^*(s)$$

Solving the last equation for $E^*(s)$ and substituting into the first gives,

$$Y^*(s) = \frac{G^*(s)R^*(s)}{1 + GH^*(s)} \quad (10.32)$$

The corresponding z-transform is $Y(z) = \dfrac{G(z)R(z)}{1 + GH(z)}$

The equation relating the inputs and outputs of Fig. 10.5 (*b*) are

$$Y(s) = M^*(s)G_2(s)\ ; \quad M(s) = E^*(s)G_1(s)\ ; \quad E(s) = R(s) - M^*(s)G_2(s)H(s)$$

Starring all equations, the solving for $Y^*(s)$ gives

$$Y^*(s) = \frac{G_1^*(s)G_2^*(s)R^*(s)}{1 + G_1^*(s)G_2H^*(s)}$$

The corresponding z-transform is $Y(z) = \dfrac{G_1(z)G_2(z)R(z)}{1 + G_1(z)G_2H(z)}$ (10.33)

Table 10.1 Z-Transform of some common functions

SI.No.	*Time function f(t), t > 0*	*z–Transform F(z)*
1.	$\delta(t)$	1
2.	$\delta(t - kT)$	z^{-k}
3.	$u_s(t) = 1,\ t \geq 0$ $= 0,\ t < 0$	$\dfrac{z}{z-1}$
4.	t	$\dfrac{Tz}{(z-1)^2}$
5.	t^2	$\dfrac{T^2 z(z+1)}{(z-1)^3}$
6.	t^{k-1}	$\lim\limits_{a \to 0} (-1)^{k-1} \dfrac{\partial^{k-1}}{\partial a^{k-1}} \left[\dfrac{z}{z - e^{-aT}}\right]$
7.	e^{-at}	$\dfrac{z}{z - e^{-aT}}$
8.	te^{-at}	$\dfrac{Tze^{-aT}}{(z - e^{-aT})^2}$
9.	$t^k e^{-at}$	$(-1)^k \dfrac{\partial^k}{\partial a^k} \left[\dfrac{z}{z - e^{-aT}}\right]$

(*Contd.*)

10.	$1 - e^{-at}$	$\dfrac{z(1-e^{-aT})}{(z-1)(z-e^{-aT})}$
11.	$\dfrac{1}{(b-a)}(e^{-at} - e^{-bt})$	$\dfrac{1}{(b-a)}\left[\dfrac{z}{z-e^{-aT}} - \dfrac{z}{z-e^{-bT}}\right]$
12.	$t - \dfrac{1}{a}(1-e^{-at})$	$\dfrac{Tz}{(z-1)^2} - \dfrac{(1-e^{-aT})z}{a(z-1)(z-e^{-aT})}$
13.	te^{-at}	$\dfrac{Tze^{-aT}}{(z-e^{-aT})^2}$
14.	$\dfrac{1}{2}\left(t^2 - \dfrac{2}{a}t + \dfrac{2}{a^2} - \dfrac{2}{a^2}e^{-at}\right)$	$\dfrac{T^2z}{(z-1)^3} + \dfrac{(aT-2)Tz}{2a(z-1)^2} + \dfrac{z}{a^2(z-1)} - \dfrac{z}{a^2(z-e^{-aT})}$
15.	$u_s(t) - (1+at)e^{-at}$	$\dfrac{z}{z-1} - \dfrac{z}{z-e^{-aT}} - \dfrac{aTe^{-aT}z}{(z-e^{-aT})^2}$
16.	$t - \dfrac{2}{a} + \left(t + \dfrac{2}{a}\right)e^{-at}$	$\dfrac{1}{a}\left[\dfrac{(aT+2)z - 2z^2}{(z-1)^2} + \dfrac{2z}{z-e^{-aT}} + \dfrac{aTe^{-aT}z}{(z-e^{-aT})^2}\right]$
17.	$\sin \omega t$	$\dfrac{z \sin \omega T}{z^2 - 2z\cos \omega T + 1}$
18.	$\cos \omega t$	$\dfrac{z(z - \cos \omega T)}{z^2 - 2z\cos \omega T + 1}$
19.	$\sinh \omega t$	$\dfrac{z \sin \omega T}{z^2 - 2z\cosh \omega T + 1}$
20.	$\cosh \omega t$	$\dfrac{z(z - \cosh \omega T)}{z^2 - 2z\cosh \omega T + 1}$
21.	$e^{-at}\sin \omega t$	$\dfrac{ze^{-aT}\sin \omega T}{z^2 - 2ze^{-aT}\cos \omega T + e^{-2aT}}$
22.	$1 - e^{-at}\sec\phi\cos(\omega t + \phi)$ $\phi = \tan^{-1}(-a/\omega)$	$\dfrac{z}{z-1} - \dfrac{z^2 - ze^{-aT}\sec\phi\cos(\omega T - \phi)}{z^2 - 2ze^{-aT}\cos \omega T + e^{-2aT}}$
23.	$e^{-at}\cos \omega t$	$\dfrac{z^2 - ze^{-aT}\cos \omega T}{z^2 - 2ze^{-aT}\cos \omega T + e^{-2aT}}$

10.7 LIMITATIONS OF THE Z-TRANSFORMATION METHOD

We should keep the following facts in mind when applying the *z*-transformation:

The sampling duration of the sampler has been assumed to be negligibly small compared to the smallest time constant of the system.

The *z*-transform Y(*z*) specifies only values of the time functions *y*(*t*) at the sampling instants. Y(*z*) does not contain any information concerning the value of y(t) between the sampling instants.

The transfer function of the continuous data system G(*s*) must have at least one more pole than zeros, which, in other words, means that the impulse response *g*(*t*) of G(*s*) must not have any jump discontinuity at $t = 0$. Otherwise, the system response obtained by the *z*-transform method will be misleading and sometimes even incorrect.

10.8 FREQUENCY DOMAIN ANALYSIS OF SAMPLING PROCESS

Let us now examine the sampling process from the point of view of frequency-domain.

The sampling process considered in the present analysis is a modulating process in which a finite amplitude pulse train *p*(*t*) with fixed period T, as shown in Fig. 10.6(*a*), is multiplied by a continuous function *e*(*t*) to produce the sampled function $e_p^*(t)$ as shown in Fig. 10.6(*b*) and Fig. 10.6(*c*) respectively. The period T is called the sampling period and is considered a fixed quantity here. However, there may be samplers (known as *pulse-width-modulator*) in which the width of the output sampled signal is a function of amplitude of the

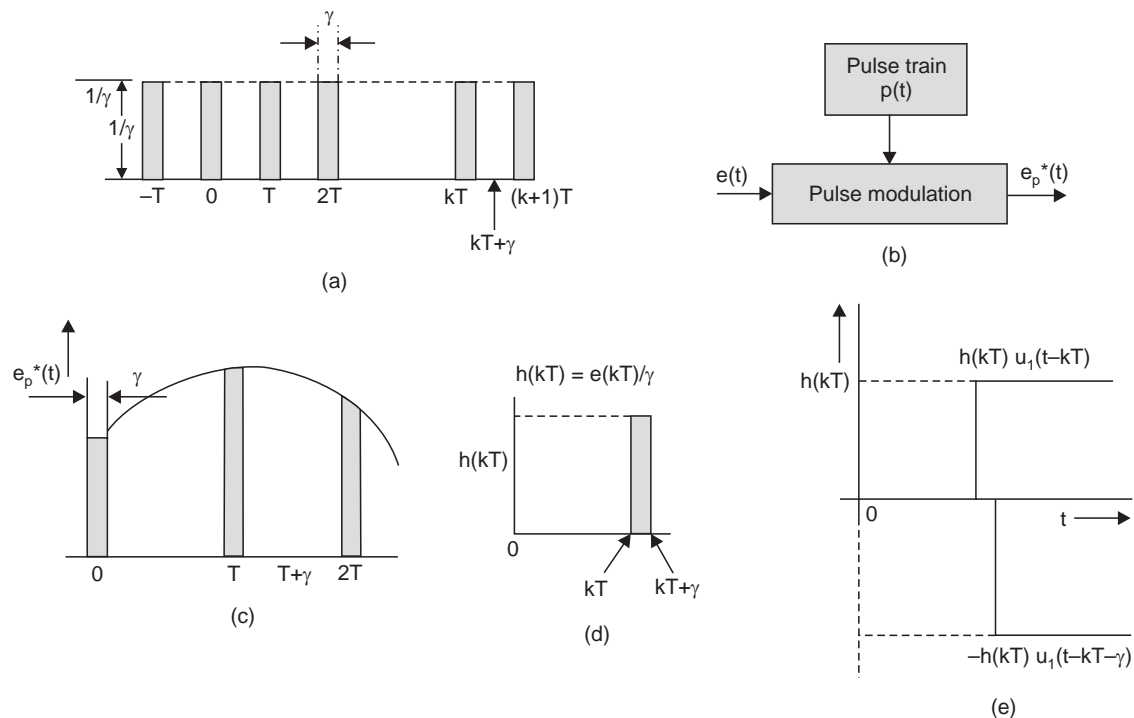



Fig. 10.6 Sampling process $0 < \gamma < T$ (*a*) Pulse train (b) Sampling device (c) Pulse signal $e_p^*(t)$ (d) pulse of height *h*(*k*T) and strength $e_p^*(t)$ (e) Mathematical equivalence of *h*(*k*T)

continuous input signal $e(t)$. There may be more complex samplers (known as *pulse-width-pulse-amplitude* samplers) in which the sampled output amplitude as well as its width is a function of the input signal at the sampling instant.

The strength of the pulse (which is the area under the pulse) is unity with height $1/\gamma$ and duration γ and period T. When the duration γ of the pulse approaches zero, the sampling process is known as impulse sampling. So, the sampled signal from the modulator output may be written as:

$$e_p^*(t) \cong p(t)\, e(t) \tag{10.34}$$

In most control situations the sensor outputs are continuous in nature, which are sampled to digital signals to exploit the processing power and flexibility of digital computers for generating control laws. Once the control law is available, it has to be converted back to continuous domain to make it compatible with the plant input which again is mostly analog in nature. In order to reconstruct a continuous signal $e(t)$ properly from its sampled version $e_p{}^*(t)$, it is necessary that the frequency content of the pulse signal contain all the frequency information initially present in $e(t)$. One way to functionally determine this frequency characteristic is to represent the pulse train $p(t)$ as a Fourier series as shown below:

$$p(t) = \sum_{m=-\infty}^{\infty} \mathrm{P}(jm\omega_s)\, e^{jm\omega_s t} \tag{10.35}$$

where $m = 0, \pm 1, \pm 2, \ldots \pm \infty$ and $\mathrm{P}(jm\omega_s)$ are defined as the Fourier co-efficients. Equation (10.35) is referred to as the complex Fourier series expansion of the pulse train $p(t)$.

Since the sampling frequency is $\omega_s = 2\pi/\mathrm{T}$, we can write

$$p(t) = \begin{cases} \dfrac{1}{\gamma} \text{ for } k\mathrm{T} \le t < k\mathrm{T} + \gamma \\ 0 \text{ for } kT + \gamma \le t < (k+1)\mathrm{T} \end{cases}$$

Then the Fourier co-efficients $\mathrm{P}(jm\omega s)$ have to be evaluated only over the time period $0 \le t < \gamma$ as shown below:

$$\mathrm{P}(jm\omega_s) = \frac{1}{\mathrm{T}} \int_0^{\gamma} p(t) e^{-jm\omega_s t}\, dt$$

or

$$\mathrm{P}(jm\omega_s) = \frac{1}{\mathrm{T}\gamma} \int_0^{\gamma} e^{-jm\omega_s t}\, dt = \frac{1 - e^{-jm\omega_s \gamma}}{\mathrm{T}\gamma\, jm\omega_s}$$

$$= \frac{2}{\mathrm{T}\gamma m\omega_s} e^{\frac{-jm\omega_s\gamma}{2}} \left[\frac{e^{\frac{jm\omega_s\gamma}{2}} - e^{\frac{-jm\omega_s\gamma}{2}}}{2j} \right] = \frac{2}{\mathrm{T}\gamma m\omega_s} \sin\left(\frac{m\omega_s\gamma}{2}\right) e^{\frac{-jm\omega_s\gamma}{2}}$$

$$= \frac{1}{\mathrm{T}} \frac{\sin\left(\dfrac{m\omega_s\gamma}{2}\right)}{\dfrac{m\omega_s\gamma}{2}} e^{\frac{-jm\omega_s\gamma}{2}} \tag{10.36}$$

Now, the Fourier transform $\mathrm{P}(j\omega)$ of $p(t)$ is defined as

$$\mathrm{P}(j\omega) = \int_{-\infty}^{\infty} p(t)\, e^{-j\omega t}\, dt \tag{10.37}$$

Substituting $p(t)$ from (10.35)

$$P(j\omega) = \sum_{m=-\infty}^{\infty} P(jm\omega_s) \int_{-\infty}^{\infty} e^{-j(-m\omega_s+\omega)t}\, dt = \sum_{m=-\infty}^{\infty} P(jm\omega_s)\, \delta(\omega - m\omega_s) \tag{10.38}$$

where $\delta(\omega - m\omega_s)$ denotes an impulse of unit strength occurring at $\omega = m\omega_s$. Hence in the frequency domain, $P(j\omega)$ is a train of impulses, the members being of strength $P(jm\omega_s)$ and occurring at $\omega = m\omega_s$, $m = 0, +1, +2, +$ ……. Thus substituting $P(jm\omega_s)$ from (10.36) into (10.38) yields the Fourier transform

$$P(j\omega) = \frac{1}{T} \sum_{m=-\infty}^{\infty} \frac{\sin\left(\frac{m\omega_s\gamma}{2}\right)}{\frac{m\omega_s\gamma}{2}} e^{\frac{-jm\omega_s\gamma}{2}} \delta(\omega - m\omega_s) \tag{10.39}$$

We can find the periodic frequency components of the sampled signal $e_p^*(t)$ by taking its Fourier transform. Since, the sampled signal, $e_p^*(t) = p(t)\, e(t)$, is the product of two time functions, we have to apply the theorems of convolution in frequency domain to find the Fourier transform. Therefore, designating the Fourier transform of $e_p^*(t)$ by $E_p^*(j\omega)$, we have

$$E_p^*(j\omega) = \int_{-\infty}^{\infty} P(j\omega)\, E[j(\omega - \omega')]\, d\omega' \tag{10.40}$$

$$= \int_{-\infty}^{\infty} \left[\sum_{m=-\infty}^{\infty} P(jm\omega_s)\, \delta(\omega' - m\omega_s) \right] E\,[j(\omega - \omega')] d\omega'$$

$$= \sum_{m=-\infty}^{\infty} P(jm\omega_s) \int_{-\infty}^{\infty} \delta(\omega' - m\omega_s)\, E\,[j(\omega - \omega')] d\omega'$$

Since the integral portion has a nonzero value when $\omega' = m\,\omega_s$ that is

$$\int_{-\infty}^{\infty} \delta(\omega' - m\omega_s)\, E\,[j(\omega - \omega')] d\omega' = E[j(\omega - \omega')]$$

We get, $$E_p^*(j\omega) = \sum_{m=-\infty}^{\infty} P(jm\omega_s)\, E[j(\omega - m\omega_s)] \tag{10.41}$$

So, using relation (10.36) for $P(jm\omega_s)$ we can write

$$E_p^*(j\omega) = \frac{1}{T} \sum_{m=-\infty}^{\infty} \frac{\sin\left(\frac{m\omega_s\gamma}{2}\right)}{\frac{m\omega_s\gamma}{2}} e^{\frac{-jm\omega_s\gamma}{2}} E\,[j(\omega - m\omega_s)] \tag{10.42}$$

Frequency response of a typical continuous function $e(t)$ is shown in Fig. 10.7 (a), which is assumed to be band limited $\omega < \omega_c$, as all practical systems are, with maximum amplitude A. The spectrum $|\,P(j\omega)|$ of the impulse, $p(t)$, is shown in part (b) of the Fig. 10.7.

The spectrum of the sampled function $E_p^*(j\omega)$ is shown in Fig. 10.7(c), where ω_s is selected equal to $2\omega_c$. It is seen that the sampling process produces periodic components. The

fundamental spectrum in part (*c*), $-\omega_c < \omega < \omega_c$ is similar in shape to that of the spectrum of the continuous function in part (*a*) of Fig. 10.7.

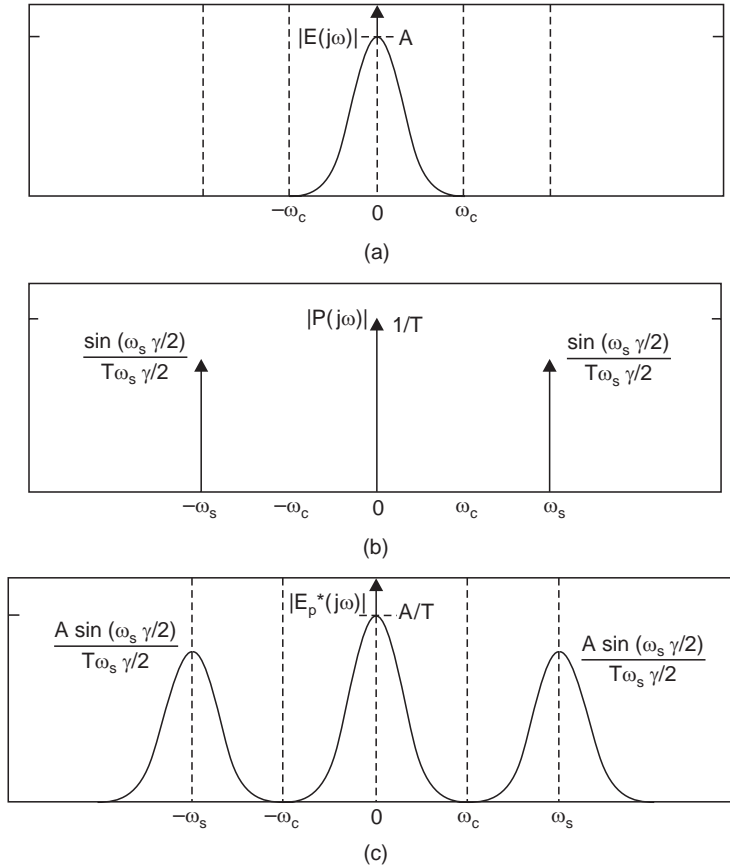



Fig 10.7 Frequency spectra for finite-pulse-width sampling frequency spectrum for $\omega_s = 2\omega_c$
(a) Band limited Frequency-spectrum of a continuous function *e*(*t*)
(b) Frequency spectra for pulse train *p*(*t*)
(c) Frequency spectra for pulse sampled function *e**(*t*)

The frequency response also produces a succession of complementary or folded spectra which are shifted periodically by a frequency separated by $m\omega_s$ ($m = \pm 1, \pm 2.....$). If the frequency is sufficiently high ($\omega_s > 2\omega_c$) there is essentially no overlap between the fundamental and complementary spectra as shown in Fig. 10.7(*c*). However, if the sampling frequency is such that $\omega_s < 2\,\omega_c$, overlap or frequency folding occurs as shown in Fig. 10.8 (*c*).

We note that the term in Equation (10.42) as $\gamma \rightarrow T$ has the value

$$\lim_{\gamma \to T} \frac{\sin\left(\dfrac{m\omega_s\gamma}{2}\right)}{\dfrac{m\omega_s\gamma}{2}} = \frac{\sin(m\,\pi)}{m\,\pi} = 0, \text{ for } m \neq 0 \qquad (10.43)$$

and all the folded components of $E^*(j\omega)$ vanish. Only fundamental components of $E_p^*(j\omega)$ remains since $[\,|\sin m\pi\,|\,/m\pi]_{m=0} = 1$. In this case the modulator in Fig. 10.6(*a*) is a continuous multiplier

of 1/T for all time. Again when $\gamma \to 0$, the pulse becomes an impulse. Under this condition, the spectrum of sampled signal becomes:

$$E_p^*(j\omega) = \lim_{\gamma \to 0} E_p^*(j\omega) = \frac{1}{T} \sum_{m=-\infty}^{\infty} E\,[j(\omega - m\omega_s)]$$

since, the limiting value $\lim_{\gamma \to 0} \dfrac{\sin\left(\dfrac{m\omega_s \gamma}{2}\right)}{\dfrac{m\omega_s \gamma}{2}} = e^{\frac{-jm\omega_s\gamma}{2}} = 1$ (by using L Hospital's Theorem) (10.44)

Therefore, the spectrum of the impulse-sampled signal becomes

$$E_p^*(j\omega) = \frac{E\,(j\omega)}{T} + \frac{E(j\omega \pm \omega_s)}{T} + \frac{E(j\omega \pm 2\omega_s)}{T} + \ldots \quad (10.45)$$

The relation (10.45) reveals that the complimentary spectra are equal to the primary spectrum and is obtained after multiplying the continuous domain spectrum with (1/T) and are separated from each other by $m\omega_s$, ($m = \pm 1, \pm 2 \ldots$). The spectrum is shown in part (*b*) and (*c*) of Fig. 10.8.

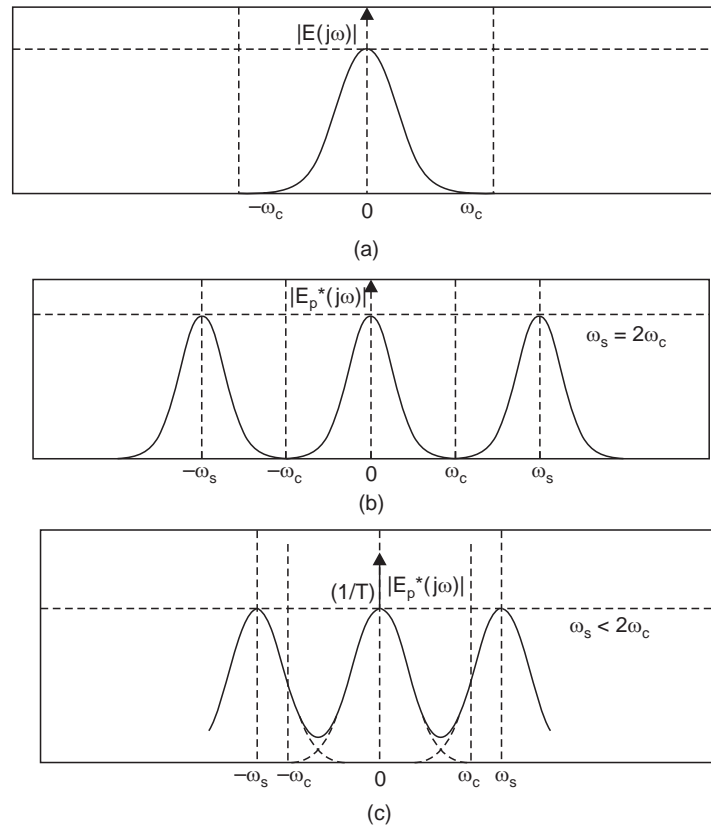



Fig. 10.8 Frequency spectra of (a) continuous function *e(t)* with highest frequency ω_c (b) an impulse-sampled function $e^*(t)$ where $\omega_s \geq 2\omega_c$ (c) an impulse -sampled function $e^*(t)$ where $\omega_s < 2\omega_c$

The spectrum in Fig. 10.9 shows the presence of alias frequencies when a continuous signal with the highest frequency ω_c rad/sec is sampled at a rate of ω_s = 100 rad/sec where $\omega_c \geq \omega_s/2$.

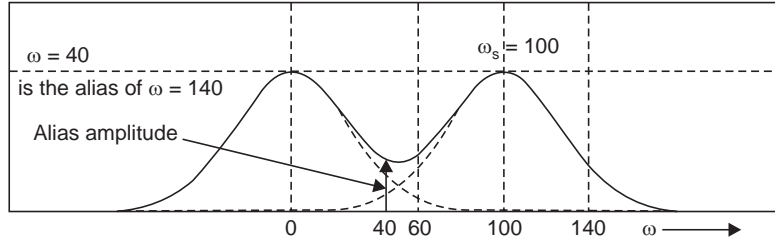



Fig. 10.9 The spectrum showing alias frequencies

In time domain, the phenomenon of aliasing has the implication that the two sinusoidal signals of different frequencies become indistinguishable after sampling—they have the same sampled value due to a particular choice of sampling rate related to the time periods of the signals. This is shown in Fig. 10.10 where two sinusoidal signals given by $e_1 = \sin(2\pi\, t/8)$ and $e_2 = \sin(18\,\pi\, t/8)$ are sampled at rate T_s = 1 sec, and the sampled signals $e^*_1(t)$ and $e^*_2(t)$ have identical values at the sampling instants t = T, 2T, 3T, ... as shown in Fig. 10.10.

We note that, since $\omega_s = 2\pi f_s = 2\pi$ rad /sec, one of the alias frequency is given by

$$\omega_3 = \omega_1 + \omega_s = 2\pi/8 + 2\pi = 18\pi/8 \text{ rad/sec}$$

and the other alias frequency is found as

$$\omega_4 = \omega_1 - \omega_s = 2\pi/8 - 2\pi = -14\pi/8$$

So, the second alias signal may be written as

$$e_2 = \sin(-14\,\pi\, t/8) = -\sin(14\pi\, t/8)$$

Since one of the alias frequency ω_3 generated by sampling the signal e_1 at a rate of $\omega_s = 2\pi$ rad/sec, is identical with the angular frequency of e_2, the sampled signal $e_2{}^*(t)$ cannot be separated from $e_1{}^*(t)$ (vide Fig. 10.10).

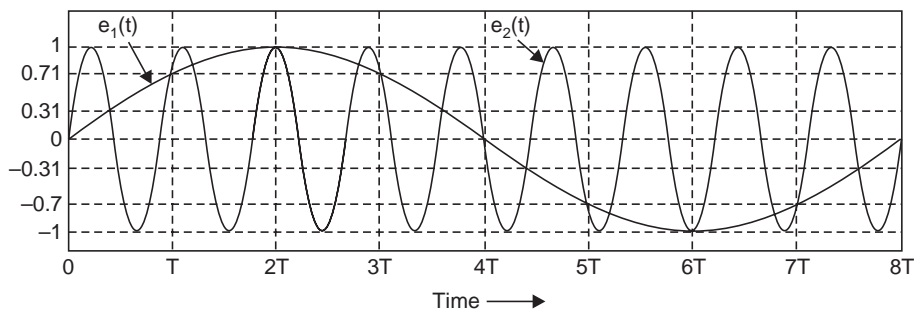



Fig. 10.10 Two sinusoids become indistinguishable when sampled at a particular rate giving rise to alias frequencies

10.9 DATA RECONSTRUCTION

In sample data control systems the microcomputer is used for the acquisition of data from sensors and process it for the generation and implementation of control laws. However, the majority of actuators and plants are analog in nature, which can not handle digital signals

Chapter 10

directly and needs continuous signals. So, in most feedback control systems employing sampled data, a continuous signal is to be reconstructed from the sampled signal. The block diagram of a simple sampled data control system is shown in Fig. 10.11.

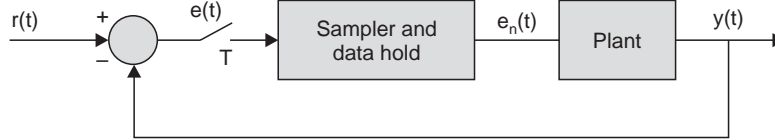



Fig. 10.11 Sampled-data control system

Suppose that the sampled signal is band limited in frequency, which is true for all practical systems, so that the highest frequency component ω_c of $e(t)$ is less than $\omega_s/2$ as in Fig. 10.12 (*a*) . Then $E^*(j\omega)$ would have the frequency spectrum shown in Fig. 10.12 (*b*) and theoretically the signal could be reconstructed exactly by employing an ideal low pass filter. However, since ideal filters do not exist in physically realizable system, we must employ approximations. Practical data holds are devices that approximate, in some sense, an ideal low pass filter.

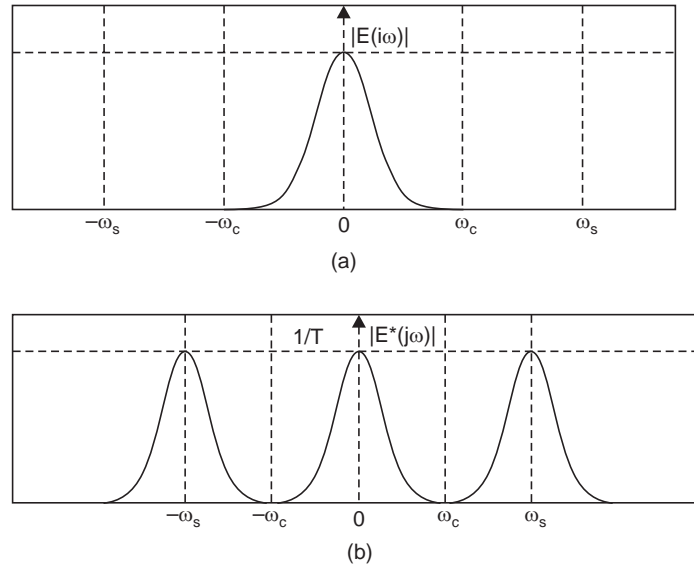



Fig. 10.12 Frequency spectrum (a) of E(jω) and (b) of E*(jω)

A commonly used method of data reconstruction, is polynomial extrapolation. Using a Taylor's expansion, about $t = n\text{T}$, we can express $e(t)$ as

$$e(t) = e(n\text{T}) + e'(n\text{T})\,(t - n\text{T}) + \frac{e''(n\text{T})}{2!}\,(t - n\text{T})^2 + \ldots \qquad (10.46)$$

where the prime denotes the derivative. The reconstructed version of $e(t)$ for the n^{th} sample period is designated as $e_n(t)$, that is,

$$e_n(t) \cong e(t) \quad \text{for} \quad n\text{T} \le t < (n + 1)\text{T} \qquad (10.47)$$

The output of the data hold will be represented by $e_n(t)$. Since $e(t)$ enters the data hold only in sampled form, the values of the derivatives can not be computed in the way it is done

for continuous signals. However, the derivatives may be approximated by the backward difference as shown below:

$$e'(nT) = \frac{1}{T}\,[e(nT) - e\,\{(n-1)\}T] \tag{10.48}$$

$$e''(nT) = \frac{1}{T}\,[e'(nT) - e'\,\{(n-1)T\}] \tag{10.49}$$

$$= \frac{1}{T}\left\{\frac{1}{T}\,[e(nT) - e\{(n-1)T\}] - \frac{1}{T}\,[e\{(n-1)T\} - e\,\{(n-2)\,T\}]\right\}$$

$$= \frac{1}{T^2}\,\{e(nT) - 2e[(n-1)T] + e[(n-2)T]\} \tag{10.50}$$

Based on the foregoing relationships, two different types of data hold, namely—the zero and first order holds will now be discussed.

10.9.1 Zero Order Hold

If only the first term in the expansion of (10.46) is used, the data held is called a zero-order hold, where the function $e(t)$ is assumed constant within the sampling interval at a value equal to that of the function at the preceding sampling instant. Therefore, for the zero-order hold, we have

$$e_n\,(t) = e(nT) \qquad nT \le t < (n+1)T.$$

It is to be noted that the reconstructed signal $e_n(t)$ needs only the current sampled signal $e(nT)$ and no other past value is to be stored or remembered. Hence the name zero order hold since its implementation needs zero memory and it is the simplest to implement. With reference to the model of a sample and hold system as shown in Fig. 10.13, the zero order hold needs only the impulse functions at the input whose strengths are to be maintained at the output for the rest of the sampling period as is apparent from the expression of $e_n(t)$.

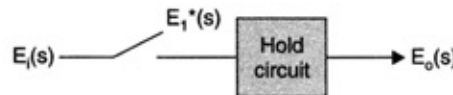



Fig. 10.13 Representation of sample and hold circuit

The signal $e_o(t)$ shown in Fig. 10.14 (b) then describes the data hold output if the input $e_i(t)$ to the data hold is a unit impulse as shown in part (a) of the fig. So, the sampler output may be written as

$$e_o(t) = u_s\,(t) - u_s\,(t - T) \tag{10.51}$$

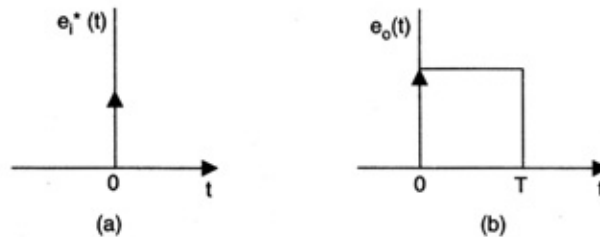



Fig. 10.14 Input and output signals of zero order hold

The Laplace transform is given by

$$E_o(s) = \frac{1}{s} - \frac{e^{-Ts}}{s} = \frac{1 - e^{-Ts}}{s}$$

Since $E_i(s) = 1$, the transfer function of the zero order hold is given by

$$G_{ho}(s) = \frac{E_0(s)}{E_i(s)} = \frac{1 - e^{-Ts}}{s} \tag{10.52}$$

The frequency response of the zero-order-hold can be obtained by replacing s by $j\omega$

$$G_{ho}(j\omega) = \frac{1 - e^{-j\omega T}}{j\omega} = \frac{2e^{-j\omega T/2}}{\omega}\left[\frac{e^{j\omega T/2} - j\omega T/2}{2j}\right]$$

$$= \frac{2e^{-j\omega T/2}}{\omega} \sin\left(\omega T/2\right) = \frac{T \,.\, \sin\,(\omega T/2)}{\omega T/2}\, e^{j\omega T/2} \tag{10.53}$$

Since,
$$\frac{\omega T}{2} = \frac{\omega}{2}\cdot\left(\frac{2\pi}{\omega_s}\right) = \frac{\pi\omega}{\omega_s}, \tag{10.54}$$

relation (10.53) can be simplified as
$$G_{ho}(j\omega) = T \,.\, \frac{\sin\left(\pi \frac{\omega}{\omega_s}\right)}{\pi \frac{\omega}{\omega_s}} \,.\, e^{-j\left(\pi \frac{\omega}{\omega_s}\right)} \tag{10.55}$$

Thus
$$|\,G_{ho}(j\omega)\,| = T \,.\left|\frac{\sin\left(\pi \frac{\omega}{\omega_s}\right)}{\pi \frac{\omega}{\omega_s}}\right| \quad \text{and} \quad \angle G_{ho}(j\omega) = -\frac{\pi\omega}{\omega_s} = \theta \tag{10.56}$$

The amplitude and phase plots for $G_{ho}(j\omega)$ are shown in Fig. 10.15. It is to be noted that the phase of $\sin\left(\pi \frac{\omega}{\omega_s}\right)$ changes from 180° to – 180° at integer values of ω/ω_s = 1, 2 .. etc, this has been taken care of in the phase plot of ZERO ORDER HOLD in Fig. 10.15(*b*).

In order to interpret the frequency response of the zero-order hold, we must remember the fact that the data hold is preceded by an ideal sampler. So, if a sinusoidal signal $E_i \sin(\omega_1 t)$ is applied to the ideal sampler, where $\omega_1 < \omega_s/2$, then the sampled output $E_i^*(j\omega)$ contains additional frequencies which were not present in the input signal as shown in Fig. 10.16(*b*). The frequency spectrum of the zero-order hold output $E^*_0(j\omega)$ can be computed by convolution of $G_{ho}(j\omega)$ with $E_i^*(j\omega)$ and is shown in Fig. 10.16(*c*).

We note that the output signal amplitude spectrum will be the same as that shown in the figure, if the input signal frequency is $\omega = k\omega_s \pm \omega_1$, k = 0, 1, 2. Hence any frequencies $\omega_1 > \omega_s/2$ will reflect into the frequency range $0 < \omega < \omega_s/2$. This effect is called frequency fold-over or frequency aliasing. These reflected frequencies will be interpreted as low-frequency information in $e(t)$, which can not be tolerated in general. The frequency aliasing can be prevented either by increasing ω_s or by placing an analog anti-aliasing low pass filter in front of the sampler to remove any frequencies in $e_i(t)$ that are greater than $\omega_s/2$. In practice, the

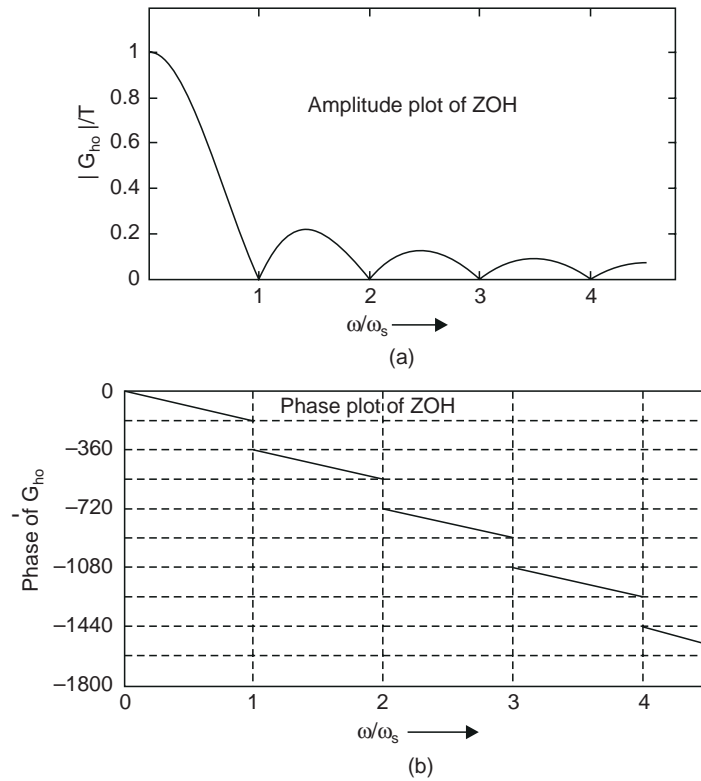



Fig. 10.15 Amplitude and phase plot of zero order hold

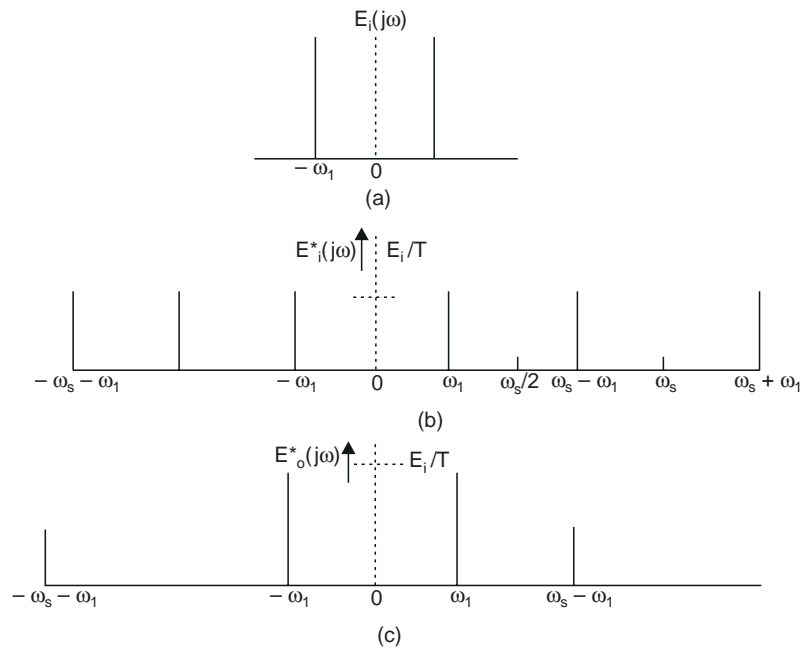



Fig. 10.16 Sinusoidal response of the sample and a zero-order hold (a) Intput signal to ideal sampler (b) Output signal of the ideal sampler (c) Output signal from zero order hold

sampling frequency is chosen at a value much higher than $2\omega_c$, normally 10 times the critical frequency ω_c in control system and 3 to 5 times in communication systems.

So, for a system with frequency response $G(j2\pi f)$ shown in Fig. 10.17, the sampling frequency f_s will be at least 3000 Hz and in that case the 400 Hz noise from the aircraft power supply along with the noise from bending moment around 50 Hz will be present in the reconstructed signal. In such a situation anti-aliasing low pass filter with cutoff frequency around 35 Hz is to be used before the sampler circuit.

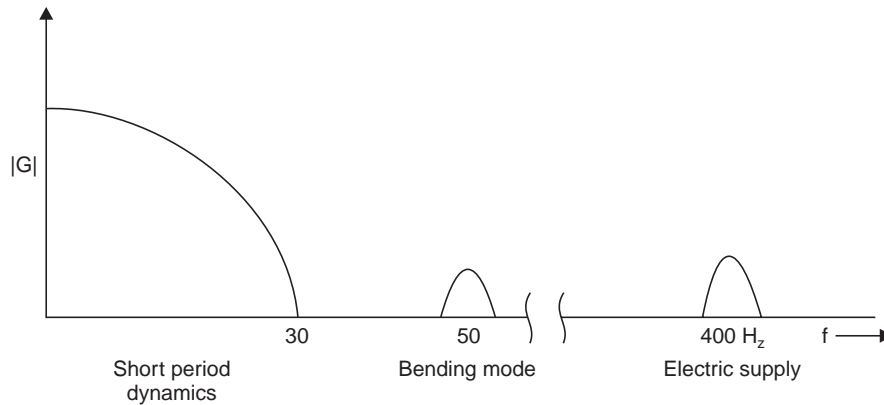



Fig. 10.17 Frequency response of short period dynamics of aircraft systems

10.10 FIRST ORDER HOLD

The first order hold is realized by retaining the first two terms of (10.46). Therefore,

$$e_n(t) = e(nT) + e'(nT)(t - nT), \quad nT \le t < (n + 1)\, T \tag{10.57}$$

where,
$$e'(nT) = \frac{e(nT) - e[(n-1)T]}{T} \tag{10.58}$$

This expression indicates that the extrapolated function within a given interval is linear and that its slope is determined by the values of the function at the sampling instants in the previous interval. We note that a memory element is required to store the value of $e[(n - 1)T]$ for the realization of this data hold. To determine the transfer function of this data hold, assume that the input is unit impulse function at $t = 0$. Then, from (10.57) and (10.58), the output is obtained as shown in Fig. 10.18 (*b*). The Fig. 10.18 (*b*) can be drawn by noting that

$$e_n(t) = e(nT) + \frac{e(nT) - e[(n-1)T]}{T}\, h \quad \text{where } 0 \le h < T$$

(*i*) with $n = 0$, $e_0(t) = e(0) + \dfrac{e(0) - 0}{T}\, h \quad \therefore \quad$ at $h = 0$, $e_0(0) = e(0)$ and $h \to T$, $e_0(T) = 2e(0)$

(*ii*) with $n = 1$, $e_1(t) = e(T) + \dfrac{e(T) - e(0)}{T}\, h \quad$ for $\quad T \le t < 2T$

$$= -\frac{h}{T}\, e(0), \text{ since } e(T) = 0$$

The plot of $e_1(t)$ is shown in Fig. 10.18 (*b*) which can be decomposed in six components as shown in part (*c*) of the figure. The components are

$$e_0(t) = u_s(t) + \frac{1}{T}t - 2u_s(t - T) - \frac{2}{T}(t - T) + u_s(t - 2T) + \frac{t - 2T}{T}$$

where $u_s(t) = \begin{cases} 1 & , t \geq 0 \\ 0 & , t < 0 \end{cases}$ and $u_s(t - T) = \begin{cases} 1 & , t \geq T \\ 0 & , t < T \end{cases}$

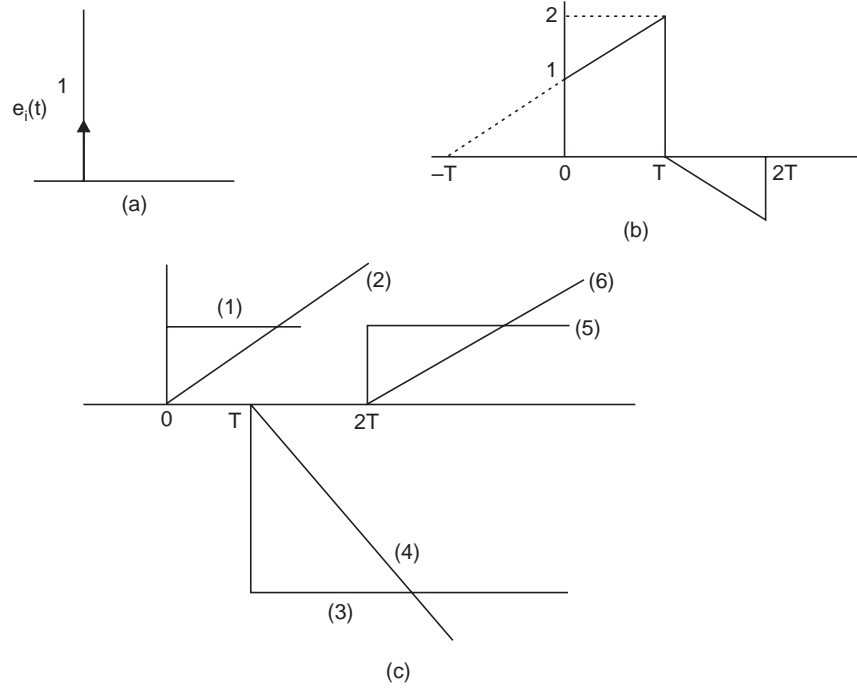



Fig. 10.18 Impulse response of first order hold circuit

Taking the Laplace transform of $e_o(t)$, we get

$$E_o(s) = \frac{1}{s} + \frac{1}{T}\cdot\frac{1}{s^2} - \frac{2e^{-Ts}}{s} - \frac{2e^{-Ts}}{Ts^2} + \frac{e^{-2Ts}}{s} + \frac{1}{T}\cdot\frac{e^{-2Ts}}{s^2}$$

$$= \frac{1}{s}(1 - e^{-Ts})^2 + \frac{1}{s^2T}[1 - e^{-Ts}]^2$$

$$= (1 - e^{-Ts})^2\left(\frac{1}{s} + \frac{1}{s^2T}\right) = \left(\frac{sT + 1}{s^2T}\right)(1 - e^{-Ts})^2$$

$$= \left(\frac{sT + 1}{T}\right)\left(\frac{1 - e^{-Ts}}{s}\right)^2 = T(1 + sT)\left(\frac{1 - e^{-Ts}}{sT}\right)^2$$

Since $E_i(s) = 1$
$$\frac{E_o(s)}{E_i(s)} = Gh_1(s) = T(1 + sT)\left[\frac{1 - e^{-Ts}}{sT}\right]^2 \qquad (10.58)$$

The frequency response of the first order hold is obtained from (10.58) as

$$G_{h1}(j\omega) = T(1 + j\omega T)\left[\frac{1 - e^{-j\omega T}}{-j\omega T}\right]^2 \tag{10.59a}$$

$$|\, G_{h1}\,(j\omega)\,| = T\sqrt{1 + \frac{4\pi^2\omega^2}{\omega_s^2}}\left[\frac{\sin\left(\frac{\pi\omega}{\omega_s}\right)}{\frac{\pi\omega}{\omega_s}}\right]^2 \tag{10.59b}$$

The phase is given by $\angle G_{h1}\,(j\omega) = \tan^{-1}\left(\frac{2\pi\omega}{\omega_s}\right) - \frac{2\pi\omega}{\omega_s} = \theta$ (10.59c)

The amplitude and phase plot of first order hold is shown in Fig. 10.19.

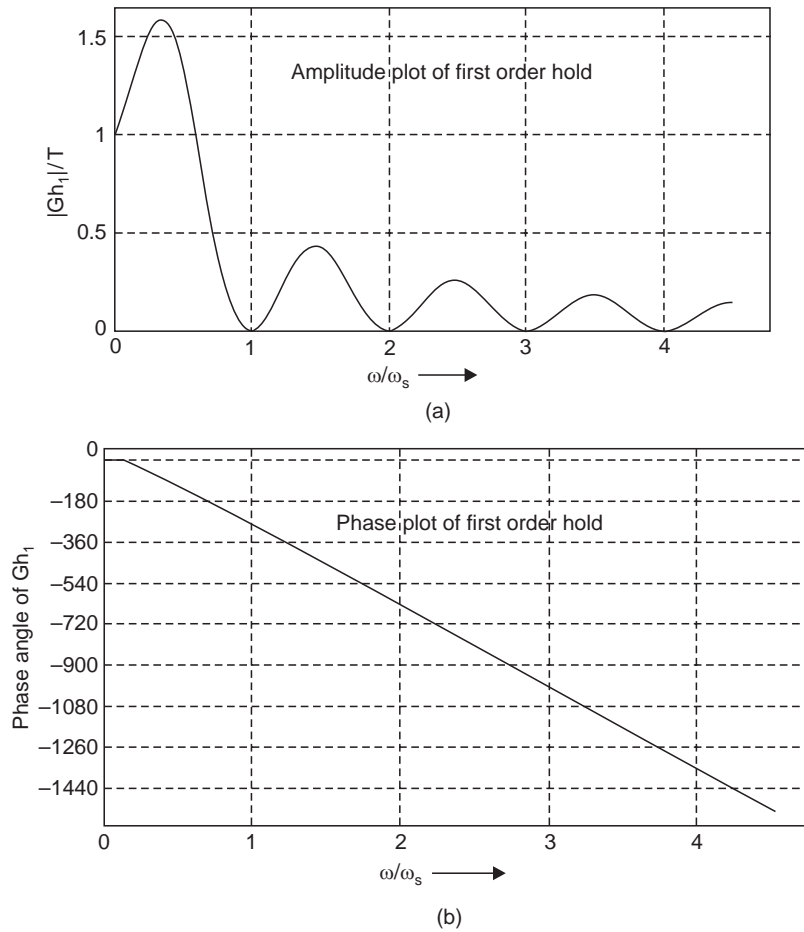



Fig. 10.19 Amplitude and phase plot first order hold circuit

If a series of impulses is impressed as input to the first order hold figure as shown in Fig. 10.20 (*a*), the corresponding output reconstructed by the first order hold is shown in Fig. 10.20 (*b*).

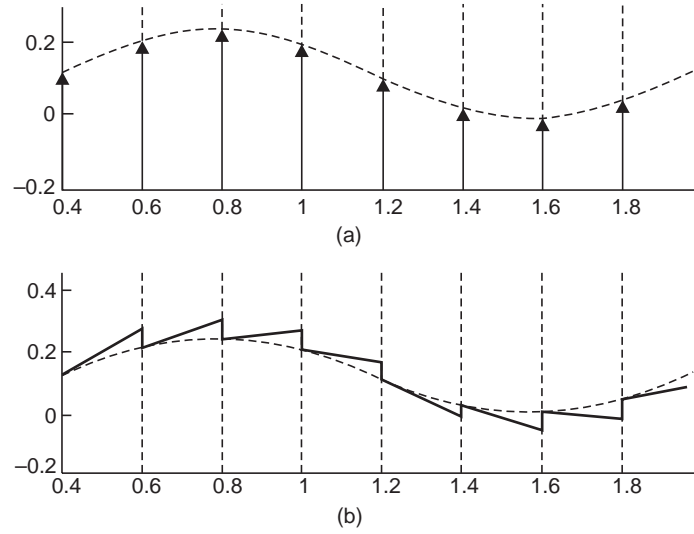



Fig. 10.20 Input and output of a first order hold data re-constructor (a) Series of impulses as input and (b) Re-constructed output

10.11 DISCRETE STATE EQUATION

We present here a technique for finding the discrete state equations of a sampled-data system directly from the continuous state equations. In this technique the states of the continuous model become the states of the discrete model. So the natural states of the analogue system are also preserved in the discrete model. In order to develop the discrete model, let us consider that the continuous plant $G_p(s)$ is preceded by a zero order hold (ZOH) circuit (vide Section 10.9.1) as shown in Fig. 10.21.

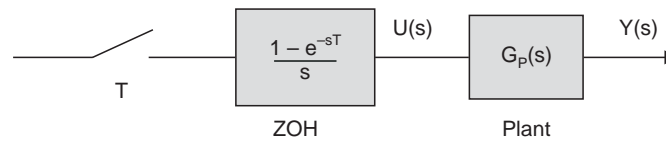



Fig. 10.21 Sampled data system

The state equation for the continuous plant $G_p(s)$ may be represented as:

$$\dot{v}(t) = A_c\, v(t) + B_c\, u(t) \tag{10.60}$$

$$y(t) = C_c\, v(t) + D_c\, u(t)$$

where $v(t)$ is the state vector, A_c is the system matrix, B_c is the input matrix, and C_c is the output matrix. The suffix c stands for the continuous system. So the solution of the system of the continuous plant can be written from (3.25) as

$$v_c(t) = e^{A_c(t-t_0)}\, v_c(t_0) + \int_{t_0}^{t} e^{A_c(t-\tau)}\, B_c\, u(\tau)\, d\tau$$

$$= \phi_c(t-t_0) v_c(t_0) + \int_{t_0}^{t} \phi_c(t-\tau)\, B_c\, u(\tau)\, d\tau \tag{10.61}$$

where the initial time is t_0 and $\phi_c(t - t_0) = e^{A_c(t-t_0)}$ is the state transition matrix. (10.62)

To obtain the discrete time model, we evaluate (10.61) at $t = (k + 1)T$ with $t_0 = kT$ yielding

$$v_c \{(k + 1)T\} = \phi_c (T) v_c (kT) + \left[\int_{kT}^{(k+1)T} \phi_c \{(k + 1)T - \tau\}B_c \, d\tau\right] m(kT) \quad (10.63)$$

We note that during the time interval $kT \le t < (k + 1)T$, the output $u(t)$ of the hold circuit, which is input to continuous plant, remains constant and is replaced by $m(kT)$. It is to be borne in mind that even though $m(kT)$ remains constant for an interval, it will change, in general, to a different constant value in the next interval depending on the nature of $u(t)$. Now we obtain the discrete state equations for the sampled data systems by rewriting Equation (10.63) in the form

$$x[(k + 1)T] = Ax(kT) + Bm(kT) \quad (10.64)$$

where $x(kT) = v_c (kT)$

$$A = \phi_c (T) = e^{A_c T} \quad (10.65)$$

and $$B = \int_{kT}^{kT+T} \phi_c (kT + T - \tau) B_c \, d\tau \quad (10.66)$$

The output equation

$$y (kT) = C_c v(kT) = C_c x(kT) \quad (10.67)$$

The Equation (10.66) may be simplified by substituting $\beta = (k + 1) T - \tau$, which implies $d\beta = - d\tau$

Therefore, $$\int_{kT}^{(k+1)T} \phi\{(k + 1)T - \tau\} \, d\tau \, B_c = - \int_{T}^{0} \phi(\beta) \, d\beta \, B_c$$

$$= \int_{0}^{T} \phi_c (\tau) \, d\tau \, B_c = \theta(T), \text{ say} \quad (10.68)$$

Equation (10.64) then becomes

$$x [(k + 1)T] = A \, x(kT) + \theta(T) \, m (kT) \quad (10.69)$$

The system matrix A in Equation (10.65) may also be evaluated as

$$A = \phi_c (T) = £^{-1} (sI - A_c)^{-1} \big|_{t = T} \quad (10.70)$$

while $\theta(t)$ may be computed as $$\theta(T) = £^{-1} \left[(sI - A_c)^{-1} \frac{B_c}{s}\right] \quad (10.71)$$

We can also compute $\theta(T)$ by evaluation of the integral in (10.68) as

$$\theta(T) = \left[\int_0^T \phi_c (\tau) \, d\tau\right] B_c = \int_0^T \left(1 + A_c \tau + A_c^2 \frac{\tau^2}{2!} + A_c^3 \frac{\tau^3}{3!} + \ldots\right) d\tau \, B_c$$

$$= \left[IT + A_c \frac{T^2}{2!} + A_c^2 \frac{T^3}{3!} + A_c^3 \frac{T^4}{4!} + \ldots\right] B_c \quad (10.72)$$

Example 10.14 For the sampled data system of Fig. 10.21 with a sampling period of T = 0.05 let $G_p(s)$ be given by:

$$G_p(s) = \frac{16}{s(s + 2)} = \frac{16}{s^2 + 2s}$$

The simulation of the plant $G_p(s)$ is shown in Fig. 10.22

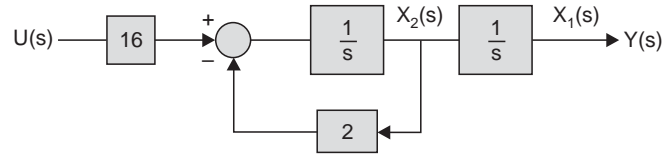



Fig. 10.22 Simulation of continuous plant $G_p(s)$ of Example 10.14

From the Fig. 10.22, we can write

$$\dot{x}(t) = \begin{bmatrix} 0 & 1 \\ 0 & -2 \end{bmatrix} x(t) + \begin{bmatrix} 0 \\ 16 \end{bmatrix} u(t) \tag{10.73}$$

$$y(t) = [1 \;\; 0]\, x(t) \tag{10.74}$$

In this case, the state transition matrix is found as

$$\phi_c(t) = £^{-1}[sI - A_c]^{-1} = £^{-1}\begin{bmatrix} s & -1 \\ 0 & s+2 \end{bmatrix}^{-1}$$

$$= £^{-1}\begin{bmatrix} s & \dfrac{1}{s(s+2)} \\ 0 & \dfrac{1}{(s+2)} \end{bmatrix} = \begin{bmatrix} 1 & \dfrac{1}{2}(1-e^{-2t}) \\ 0 & e^{-2t} \end{bmatrix} = e^{At}$$

also $$\int_0^T \phi_c(T)\, dt = \begin{bmatrix} T & \dfrac{1}{4}(2T - 1 + e^{-2T}) \\ 0 & \dfrac{1}{2}(1-e^{-2T}) \end{bmatrix}$$

Then $$e^{AT} = \phi_c(T)\,|_{T=0.05} \begin{bmatrix} 1 & 0.04758 \\ 0 & 0.90484 \end{bmatrix}$$

and $$B = \theta(T) = \left[\int_0^T \phi_c(\tau)\, d\tau\right] b_c = \begin{bmatrix} 0.05 & 0.00121 \\ 0 & 0.04758 \end{bmatrix}\begin{bmatrix} 0 \\ 16 \end{bmatrix} = \begin{bmatrix} 0.01935 \\ 0.76130 \end{bmatrix}$$

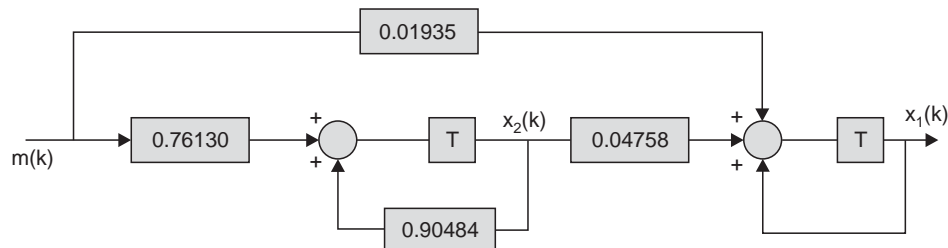



Fig. 10.23 Simulation diagram for discrete version of $G_p(s)$ in Example 10.14

Hence, the discrete state equations are

$$x(k+1) = \begin{bmatrix} 1 & 0.04758 \\ 0 & 0.90484 \end{bmatrix} x(k) + \begin{bmatrix} 0.01935 \\ 0.76130 \end{bmatrix} m(k) \tag{10.75}$$

$$y(k) = [1 \quad 0]\, x(k) \tag{10.76}$$

A simulation diagram of this model is shown in Fig. 10.23, where T represents the delay of one sampling period. Even though the states, the input and the output of the two diagrams in Figures 10.22 and 10.23 are equal at the sampling instants, the two diagrams bear no resemblance to each other. In general, the two simulation diagrams for such a system are not similar.

The discrete matrices in the above example may also be calculated by computer using relations (10.72), where A_c is found from (10.73). Five terms in the series expansion of $\phi_c(t)$ are required to yield six significant-figure accuracy. The discrete system in Equations (10.75) and (10.76) can be obtained by entering the following in the MATLAB command window:

```
>> Ac=[0 1;0 -2];
>>bc=[0;16];
>> cc = [10];
>> dc=0;
>>T=0.05
>>con_sys= ss(Ac, bc, cc, dc);
>>dis_sys=c2d(Ac, bc, cc, dc, T)
```

% The command *ss*(A*c*, *bc*, *cc*, *dc*) creates a continuous-time state-space (*ss*) model *con_sys* with matrices *Ac, bc, cc, dc* and the command *c2d(Ac, bc, cc, dc, T)* converts continuous-time models to discrete time model with sampling period *T assuming a Zero Order Hold* by default.

10.12 STATE EQUATIONS OF SYSTEMS WITH DIGITAL COMPONENTS

When the components constituting the system are all digital in nature and the inputs and outputs of the system are also in digital form, the system may be described by the following discrete dynamic equations:

$$x\,[(k+1)T] = Ax(kT) + Bu(kT) \tag{10.77}$$

$$y(kT) = Cx(kT) + Du(kT) \tag{10.78}$$

The form of Equations (10.77) and (10.78) are similar to that in Equations (10.69) and (10.67) respectively, however, they are not identical. The matrix A in Equation (10.69) is the state transition matrix $\phi(kT)$ of the system matrix A_c of the continuous system, and $\theta(kT)$ is a function of $\phi(t)$ and the input matrix B of the continuous system; thus they must all conform with the properties of the state transition matrix. The matrices A and B in Equations (10.77) and (10.78), however, are dependent entirely on the characteristics of the digital system.

10.13 THE SOLUTION OF DISCRETE STATE EQUATIONS

We shall now consider the solutions of the discrete state Equations (10.69) and (10.77) respectively in order to compute the corresponding outputs $y(kT)$ from Equations (10.67) and (10.78) respectively. Let us take up the case of the discrete system derived from continuous system and for ready reference we reproduce below Equations (10.67) and (10.69) respectively

$$y\,(kT) = C_c\, v(kT) = C_c\, x(kT)$$

$$x\,[(k+1)T\,] = A\, x(kT) + \theta\,(T)\, m\,(kT)$$

10.13.1 The Recursive Method

The most straightforward method of solving Equation (10.69) is by recursion. The following equations are written by substituting $k = 0, 1, 2$ ….. in Equation (10.69) successively.

$k = 0, \quad x(\mathrm{T}) = \mathrm{A}\, x(0) + \theta(\mathrm{T})\, m(0)$ (10.79)

$k = 1, \quad x(2\mathrm{T}) = \mathrm{A}\, x(\mathrm{T}) + \theta(\mathrm{T})\, m(\mathrm{T})$

Substituting the value of $x(\mathrm{T})$ from (10.79), we have

$$x(2\mathrm{T}) = \mathrm{A}[\mathrm{A}\, x(0) + \theta(\mathrm{T})\, m(0)] + \theta(\mathrm{T})\, m(\mathrm{T})$$

$$= \mathrm{A}^2 x(0) + \mathrm{A}\theta(\mathrm{T})\, m(0) + \theta(\mathrm{T})\, m(\mathrm{T}) \tag{10.80}$$

$k = 2, \quad x(3\mathrm{T}) = \mathrm{A}\, x(2\mathrm{T}) + \theta(\mathrm{T})\, m(2\mathrm{T})$

Using the values of X(2T) from (10.80),

$$x(3\mathrm{T}) = \mathrm{A}^3 x(0) + \mathrm{A}^2 \theta(\mathrm{T})\, m(0) + \mathrm{A}\,\theta(\mathrm{T})\, m(\mathrm{T}) + \theta(\mathrm{T})\, m(2\mathrm{T})$$

So we can write, in general, for any $k = n \geq 0$

$$x(n\mathrm{T}) = \mathrm{A}^{(n)} x(0) + \sum_{i=0}^{n-1} \mathrm{A}^{(n-i-1)}\, \theta(\mathrm{T}) m(i\mathrm{T}) \tag{10.81}$$

$$x(n\mathrm{T}) = \varphi(n\mathrm{T}) x(0) + \sum_{i=0}^{n-1} \varphi[(n-i-1)\mathrm{T}]\, \theta(\mathrm{T})\, m(i\mathrm{T}) \tag{10.82}$$

which is the solution of Equation (10.69) given the initial state $x(0)$ and the input $m(i\mathrm{T})$, for $i = 0, 1, \ldots\, n - 1$. In deriving Equation (10.82) we have used the property in (3.84)

$$\phi(\mathrm{T})\phi(\mathrm{T}) = \phi^2(\mathrm{T}) = \phi(2\mathrm{T}) \text{ and so on where } \mathrm{A} = \phi(\mathrm{T}) = e^{\mathrm{A}_c \mathrm{T}}$$

The output at the n-th sampling instant can, therefore, be written as

$$y(n\mathrm{T}) = \mathrm{C}_c\, x(n\mathrm{T}) = \mathrm{C}_c\, x(n\mathrm{T}) = \mathrm{C}_c\, [\phi(n\mathrm{T}) x(0) + \sum_{i=0}^{n-1} \phi\, [(n-i-1)\mathrm{T}] \theta(\mathrm{T})\, m(i\mathrm{T})] \tag{10.83}$$

The solutions of Equation (10.77) can be obtained recursively as shown below:

For $\quad k = 0,\ x(\mathrm{T}) = \mathrm{A}\, x(0) + \mathrm{B}u(0)$

$k = 1,\ x(\mathrm{T}) = \mathrm{A}\, x(\mathrm{T}) + \mathrm{B}u(\mathrm{T}) = \mathrm{A}\{\mathrm{A}\, x(0) + \mathrm{B}u(0)\} + \mathrm{B}u(\mathrm{T})$

$= \mathrm{A}^2 x(0) + \mathrm{AB}u(0) + \mathrm{B}u(\mathrm{T})$

And, in general, for $k = n \geq 0$,

$$x(n\mathrm{T}) = \mathrm{A}^n x(0) + \sum_{i=0}^{n-1} \mathrm{A}^{(n-i-1)}\, \mathrm{B}u(i\mathrm{T}) \tag{10.84}$$

Consequently, $y(n\mathrm{T})$ is computed from (10.78) and (10.84) as

$$y(n\mathrm{T}) = \mathrm{C}x(n\mathrm{T}) + \mathrm{D}u(n\mathrm{T}) = \mathrm{C}\left[\mathrm{A}^n\, x(0) + \sum_{i=0}^{n-1} \mathrm{A}^{(n-i-1)}\, \mathrm{B}u(i\mathrm{T})\right] + \mathrm{D}u(n\mathrm{T}) \tag{10.85}$$

So the value of the states as well as the output at any future sampling instant n can be computed in terms of the initial state $x(0)$ and the input $u(i\mathrm{T})$, for $i = 0, 1, \ldots, n - 1$ by using relations (10.84) and (10.85) respectively.

It may be mentioned here that the solutions of the state equations given above may also be computed by the z-Transform method, where the z-transform of $x(k\mathrm{T})$ and $\phi(k\mathrm{T})$ are defined by, respectively (vide relation 10.3)

$$\mathrm{X}(z) = \sum_{k=0}^{\infty} x(k\mathrm{T}) z^{-k}$$

and $$\phi(z) = \sum_{k=0}^{\infty} \phi(kT)\, z^{-k}$$

Example 10.15 With reference to the block diagram of the system in Fig. 10.21 let the plant transfer function be given by

$$G_p(s) = \frac{1}{(s+1)(s+2)} = \frac{1}{s^2+3s+2}$$

We are interested to illustrate the analysis of an open-loop discrete-data system by the state-variable method presented above. The state variable representation of the open loop plant can be written as:

$$\begin{bmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t)$$

$$y(t) = x_1(t)$$

where $x_1(t)$ and $x_2(t)$ are the state variables, $y(t)$ is the scalar output, and $u(t)$ is the scalar input. Also, since $u(t)$ is the output of the zero-order hold,

$$u(t) = u(kT) = m(kT), \text{ for } kT \le t < (k+1)T.$$

We have in this case

$$A_c = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix},\ b_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{10.86}$$

Therefore, the resolvent matrix is computed as

$$\phi(s) = (sI - A_c)^{-1} = \begin{bmatrix} s & -1 \\ 2 & s+3 \end{bmatrix}^{-1} = \frac{1}{s^2+3s+2} \begin{bmatrix} s+3 & 1 \\ -2 & s \end{bmatrix} \tag{10.87}$$

The state transition matrix $\phi(t)$ is obtained by taking the inverse Laplace transform of $\phi(s)$.

Therefore, $\phi(t)$ is computed from (10.87) as

$$\phi(t) = \begin{bmatrix} 2e^{-t} - e^{-2t} & e^{-t} - e^{-2t} \\ -2e^{-t} + 2e^{-2t} & -e^{-t} + 2e^{-2t} \end{bmatrix} \tag{10.88}$$

Substitution of b_c from (10.86) and $\phi(t)$ from (10.88) in Equation (10.68) yields

$$\theta(T) = \int_0^T \phi(T-\tau)\, b_c\, d\tau$$

$$= \int_0^T \begin{bmatrix} e^{-(T-\tau)} - e^{-2(T-\tau)} \\ -e^{-(T-\tau)} + 2e^{-2(T-\tau)} \end{bmatrix} d\tau = \begin{bmatrix} 0.5 - e^{-T} + 0.5\, e^{-2T} \\ e^{-T} - 2e^{-2T} \end{bmatrix} \tag{10.89}$$

Now substituting $t = T$ in $\phi(t)$, the discrete state equation can be written from (10.69) as

$$\begin{bmatrix} x_1[(k+1)T] \\ x_2[(k+1)T] \end{bmatrix} = \begin{bmatrix} 2e^{-T} - e^{-2T} & e^{-T} - e^{-2T} \\ -2e^{-T} + 2e^{-2T} & -e^{-T} + 2e^{-2T} \end{bmatrix} \begin{bmatrix} x_1(kT) \\ x_2(kT) \end{bmatrix}$$

$$+ \begin{bmatrix} 0.5 - e^{-T} + 0.5\, e^{-2T} \\ e^{-T} - 2e^{-2T} \end{bmatrix} m\,(kT) \tag{10.90}$$

Using Equation (10.82), the solution of Equation (10.90) is given by

$$\begin{bmatrix} x_1(nT) \\ x_2(nT) \end{bmatrix} = \begin{bmatrix} 2e^{-nT} - e^{-2nT} & e^{-nT} - e^{-2nT} \\ -2e^{-nT} + 2e^{-2nT} & -e^{-nT} + 2e^{-2nT} \end{bmatrix} \begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix}$$

$$+ \sum_{k=0}^{n-1} \begin{bmatrix} (1-e^{-T})\, e^{-(n-k-1)T} - 0.5\,(1-e^{-2T})\, e^{-2(n-k-1)T} \\ (1-e^{-2T})\, e^{-2(n-k-1)T} - (1-e^{-T})\, e^{-(n-k-1)T} \end{bmatrix} m(kT) \quad (10.91)$$

where n is any positive integer.

If the input to the zero order hold circuit is unity step, $m(kT) = 1$, for $k = 1, 2 \ldots n$.

10.14 STABILITY OF DISCRETE LINEAR SYSTEMS

The Routh-Hurwitz criteria can not be directly applied to test stability of linear discrete systems because of the nonlinear mappings between the roots in the s and z domains. It was found that a linear continuous system is unstable if any root of the characteristic equation is in the right half of the s-plane. This right half plane may be designated by $\sigma + j\omega$ in which $\sigma > 0$. The corresponding portion of the z plane is

$$z = e^{sT} = e^{\sigma T}\, e^{j\omega T}$$

The magnitude is $\quad |\, z \,| = e^{\sigma T}$.

Therefore, $\sigma > 0$ represents points in the right hand side of the s plane for which we get, $|\, z \,| > 1$.

So the right half of the s-plane corresponds to the outside of unit circle of the z-plane as is illustrated in Fig. 10.24. Thus for stability all the roots of the z-transformed characteristic equation must lie within the unit circle in the z-plane.

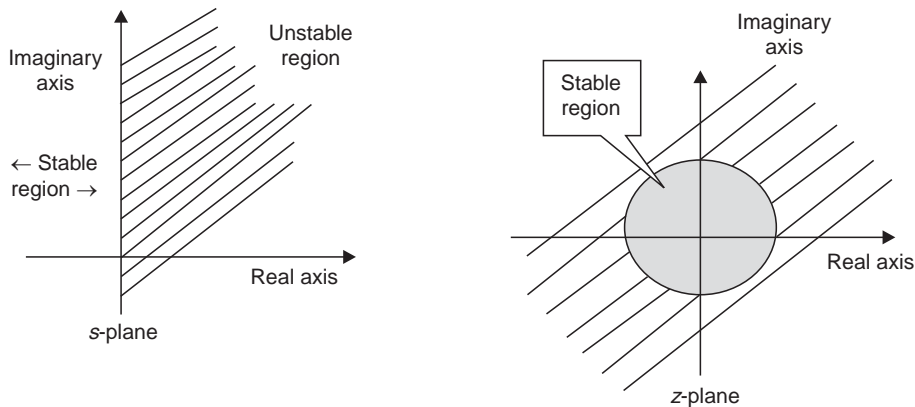



Fig. 10.24 Stability regions for s-plane and z-plane

However, we can employ a Bilinear Transformation to map the points in the z plane to the s-plane and then apply the Routh-Hurwitz criteria to the transformed characteristic equation for investigating the nature of the roots.

Let $$z = x + jy \text{ such that } x^2 + y^2 = z^2 \quad (10.92)$$

Now consider the following Bilinear Transformation:

$$z = \frac{1+\lambda}{1-\lambda} \quad (10.93)$$

such that $$\lambda = \frac{z-1}{z+1} = \frac{(x-1+jy)}{(x+1+jy)}$$

Rationalizing we get $$\lambda = \frac{(x^2+y^2)-1}{(x+1)^2+y^2} + j\,\frac{2y}{(x+1)^2+y^2} = u + jv$$

where $$u = \frac{(x^2+y^2)-1}{(x+1)^2+y^2} \text{ and } v = \frac{2y}{(x+1)^2+y^2}$$

Therefore, in view of (10.92), $z > 1$ implies $u > 0$ and $z < 1$ implies $u < 0$. That is, area inside the unity circle in the z-domain is the stability zone in λ-domain, so we can employ the bilinear relation of (10.93) to transform the characteristic equation in z domain to λ domain and apply the Routh-Hurwitz criteria to investigate stability.

Example 10.16 Consider now the system shown in Fig. 10.25 (*a*).

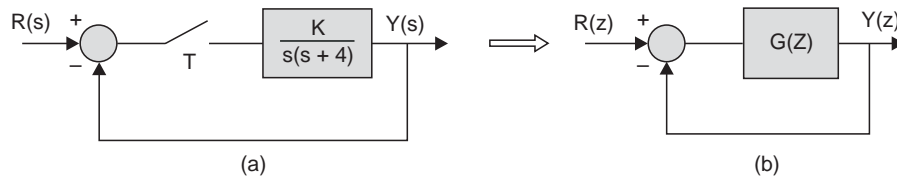



Fig. 10.25 Closed loop control system for Example 10.16

The partial fraction of the transfer function G(s) for the continuous plant is given by:

$$G(s) = \frac{K}{s(s+4)} = \frac{K}{4}\left[\frac{1}{s} - \frac{1}{s+4}\right]$$

The corresponding z-transform is

$$G(z) = \frac{K}{4}\left(\frac{z}{z-1} - \frac{z}{z-e^{-4T}}\right) = \frac{z(1-e^{-4T})\,\frac{K}{4}}{(z-1)(z-e^{-4T})}$$

Therefore, the z-transform equivalent system is shown in Fig. 10.25 (*b*)

The z-transform for the output Y(z) in Fig. 10.25 (*b*) is given by

$$Y(z) = \frac{G(z)R(z)}{1+G(z)} \tag{10.94}$$

Thus the characteristic equation for this system is obtained by setting 1 + G(z) = 0. Hence characteristic equation is

$$(z-1)\,(z-e^{-4T}) + z\,(1-e^{-4T})\,K/4 = 0 \tag{10.95}$$

In order to apply the Routh-Hurwitz criterion, it is necessary to transform the unit circle of the z-plane to the vertical imaginary axis of the λ plane. Using the relation (10.93), the characteristic equation is expressed in terms of λ, then Routh's Criterion may be applied in the same manners as for continuous systems.

For a sampling period $T = \frac{1}{4}$ sec, let us determine the value of K such that the system in Fig. 10.25 becomes unstable.

With $T = \frac{1}{4}$, The characteristic Equation (10.95) becomes

$$(z - 1)(z - 0.36788) + 0.15803\,Kz = 0 \tag{10.96}$$

The characteristic Equation (10.96) can be transformed from the z-plane to the λ-plane as:

$$\frac{0.15803\,K\,\lambda^2 + 1.26424\lambda + (2.73576 - 0.15803\,K)}{(\lambda - 1)^2} = 0.$$

The Routh array for the numerator is

λ^2	0.15803 K	2.73576 – 0.15803 K
λ^1	1.26424	0
λ^0	2.736 – 0.158 K	0

Thus the system is unstable for $K \geq 2.73576 / .15803 = 17.3116$.

Note that the continuous system (obtained by eliminating the sampler with period T in Fig. 10.25 (*a*) is stable for all values of $K > 0$.

If we use a smaller sampling period T = 0.1 sec, the characteristic Equation (10.95) becomes:

$$(z - 1)(z - 0.67032) + 0.08242\,Kz = 0$$

$$\Rightarrow \quad 0.08242\,K\lambda^2 + 0.65936\lambda + (3.34064 - 0.08242\,K) = 0$$

For stability $\quad K < 40.5319$

So we note that the range of K can be increased before the system becomes unstable by decreasing the sampling period (increasing sampling frequency).

10.14.1 Jury's Stability Test

Even though the stability of a discrete system can be tested indirectly by Routh-Hurwitz test through bilinear transformation of the characteristic equation, it is convenient to use some direct methods to accomplish the task.

Schur-Cohn criterion [30] is one of the first direct methods developed for testing the stability of a system by examining the roots of its characteristic equation with respect to the unit circle in the z-plane. It gives both the necessary and sufficient conditions for the roots to lie inside the unit circle in terms of the signs of the *Schur-Cohn determinants*. However, it is cumbersome to apply *Schur-Cohn criterion* to characteristic equation of systems of order higher than second. So, we shall rather present here an alternative and simpler method known as *Jury's stability test* [31] . In order to introduce *Jury's stability test,* let us consider an nth-order characteristic equation F(z) in z of the form

$$F(z) = a_n z^n + a_{n-1} z^{n-1} + \dots + a_2 z^2 + a_1 z + a_0 = 0 \tag{10.97}$$

where $a_0, a_1, \dots, a_n$ are real coefficients with $a_n > 0$. We arrange the coefficients of the characteristic equation F(z) in tabular form as shown in Table 10.5.1.

The first row of the table is formed by entering the coefficients of the polynomial $F(z)$ in increasing power of z starting with z^0. The entries in the second row are the elements of the first row written in reverse order. The elements of the other rows are derived in terms of the elements of these two rows.

Total number of rows in the table is $(2n - 3)$, for $n \geq 2$, where n is the order of the system.

Table 10.1

Row	z^0	z^1	z_2	...	z^{n-k}	...	z^{n-1}	z^n
1	a_0	a_1	a_2	...	a_{n-k}	...	a_{n-1}	a_n
2	a_n	a_{n-1}	a_{n-2}	...	a_k	...	a_1	a_0
3	b_0	b_1	b_2	...	b_{n-k-1}	...	b_{n-1}	
4	b_{n-1}	b_{n-2}	b_{n-3}	...	b_k	...	b_0	
5	c_0	c_1	c_2	...		c_{n-2}		
6	c_{n-2}	c_{n-3}	x_{n-4}	...		c_0		
⋮	⋮	⋮	⋮	... ...				
$2n-5$	p_0	p_1	p_2	p_3				
$2n-4$	p_3	p_2	p_1	p_o				
$2n-3$	q_0	q_1	q_2					

The elements of the table in the even $(2k + 2)^{th}$ rows, $k = 0, 1, 2,\ldots$ are the elements of the odd $(2k + 1)^{th}$ rows written in reverse order as was done for writing the elements of the second row from those in the first row. The designated elements in the odd rows in the table starting from third rows onward are formed as follows:

$$b_k = \begin{vmatrix} a_0 & a_{n-k} \\ a_n & a_k \end{vmatrix},\ c_k = \begin{vmatrix} b_0 & b_{n-k-1} \\ b_{n-1} & b_k \end{vmatrix}$$

$$d_k = \begin{vmatrix} c_0 & c_{n-2-k} \\ c_{n-2} & c_k \end{vmatrix} \tag{10.98}$$

$$q_0 = \begin{vmatrix} p_0 & p_3 \\ p_3 & p_0 \end{vmatrix},\ q_1 = \begin{vmatrix} p_0 & p_2 \\ p_3 & p_1 \end{vmatrix}$$

The necessary and sufficient conditions that none of the roots of the characteristic equation F(z) = 0 will lie on or outside the unit circle in the z-plane are:

$$F(1) > 0$$

$$F(-1) \begin{cases} > 0 & n \text{ even} \\ < 0 & n \text{ odd} \end{cases} \quad \text{or equivalently, } (-1)^n F(-1) > 0, \text{ for even or odd } n > 0$$

$$\begin{aligned} &|a_0| < a_n \\ &|b_0| > |b_{n-1}| \\ &|c_0| > |c_{n-2}| \\ &|d_0| > |d_{n-3}| \\ &\quad\vdots \qquad\quad \vdots \\ &|q_0| > |q_2| \end{aligned} \tag{10.99}$$

For a second-order system with $n = 2$, the Jury tabulation contains only one row. Therefore, for the second order system, the conditions in relation (10.99) become:

$$F(1) > 0 \tag{10.100}$$
$$F(-1) > 0$$
$$|a_0| < a_2$$

We notice from relation(10.99) that there are a total of n + 1 constraints out of which the constraints on F(1), F(– 1), and between a_o and a_n are the necessary conditions of stability. For a second-order system, the conditions in Equation (10.100) are both necessary and sufficient. It is very easy to check the necessary conditions for systems of any order, that too without preparing the Table. When the necessary conditions are met, the elaborate arrangement for preparing Jury's table may be undertaken for testing the remaining $n - 2$ constraints which are quite tedious when the characteristic equation contains variable parameters.

Some variants of Jury's tabulation, like the Raible's tabulation [32] was developed, but these do not mitigate the difficulties associated with the inequalities with variable parameters.

As in the Routh-Hurwitz criterion, occasionally the Jury test may encounter singular cases that would require remedial measures to complete the tabulation.

Singular Cases

When some of the elements of a row in the Jury tabulation are zero, Jury's table cannot be completed giving rise to a situation referred to as a *singular case.* The singular case arises when one or more roots of the characteristic equation lie on the unity circle. So, it can be handled by slightly changing the radius of the circle for moving the zeros of F(z) slightly off the unit circle $|z| = 1$. This can be done by the transformation

$$z = (1 + \varepsilon) z \tag{10.101}$$

where ε is a very small real number. The radius of the unit circle is expanded to 1 + ε, when ε is a positive number, whereas it is contracted to 1 – ε, when ε is a negative quantity. Since, ε is a very small quantity, positive or negative, the transformation in relation (10.101) can be easily applied to the characteristic equation by simply replacing the coefficient of z^n by $(1 + n\varepsilon)$ in view of the observation:

$$(1 + \varepsilon)^n z^n \cong (1 + n\varepsilon) z^n \tag{10.102}$$

The number of zeros on the unit circle can thus be found by finding the number of zeros lying inside the contracted radius and the number of zeros lying outside the expanded radius-the difference gives the number of zeros lying on the unit circle.

Example 10.17 Consider the characteristic equation given below:

$$F(z) = z^2 - 0.1z - 0.2 = (z - 0.5)(z + 0.4) = 0 \tag{10.103}$$

With reference to Equation (10.97), the coefficients of the equation are

$$a_2 = 1 \qquad a_1 = -0.1 \qquad a_0 = -0.2 \tag{10.104}$$

Since the equation is of the second order, the conditions in Equation (10.100) are necessary and sufficient for the system to be stable. Thus,

$$F(1) = 1 - 0.1 - 0.2 = 0.7 > 0 \tag{10.105}$$
$$F(-1) = 1 + 0.1 - 0.2 = 0.9 > 0 \tag{10.106}$$
$$|a_0| = 0.2 \text{ and } a_2 = 1, \text{ hence } |a_0| < a_2 \tag{10.107}$$

Thus all the conditions of (10.100) are satisfied, so all the roots are within unit circle and hence the system is stable as expected.

Example 10.18 Consider the following characteristic equation:

$$F(z) = z^3 - 1.9z^2 - 0.4z + 0.4 = 0 \tag{10.108}$$

We have $a_3 = 1 \quad a_2 = -1.9 \quad a_1 = -0.4 \quad a_0 = 0.4$ (10.109)

Checking the first three conditions in Equation (10.99),

$$F(1) = 1 - 1.9 - 0.4 + 0.4 = -0.9 \qquad (10.110)$$

$$F(-1) = -1 - 1.9 + 0.4 + 0.4 = -2.1 \qquad (10.111)$$

$$|a_0| = 0.4 < 1 = a_3 \qquad (10.112)$$

Since F(1) is negative, the necessary conditions are not satisfied, so all the roots of Equation (10.108) are not inside the unit circle, and the system is unstable. There is no need to carry out the Jury tabulation. As a matter of fact F(z) can be factored as

$$F(z)= (z - 2)(z + 0.5)(z - 0.4).$$

Example 10.19 Consider the following characteristic equation of a discrete-data system:

$$z^3 - 2.200z^2 + 1.2100z + 0.3380 = 0 \qquad (10.113)$$

We note that the equation has roots at $z = -0.2$, $z = 1.2 + j0.5$, and $z = 1.2 - j0.5$ and hence, the system is unstable.

Now applying the necessary conditions of the Jury test in Equation (10.99), we have

$$F(1) = 0.3480 \qquad (10.114)$$

$$F(-1) = -4.0720 \qquad (10.115)$$

$$|a_o| = 0.3389 < 1, \text{ so } |a_0| < a_3 \qquad (10.116)$$

Thus, we must carry out the Jury tabulation as follows to determine the stability of the system.

Row	z^0	z^1	z^2	z^3
1	0.3380	1.2100	– 2.200	1.0000
2	1.0000	– 2.200	1.2100	0.3380
3	b_0	b_1	b_2	

where $b_0 = \begin{vmatrix} a_0 & a_3 \\ a_3 & a_0 \end{vmatrix} = a_0^2 - a_3^2 = -0.8858$ (10.117)

$$b_1 = \begin{vmatrix} a_0 & a_2 \\ a_3 & a_1 \end{vmatrix} = a_0 a_1 - a_2 a_3 = 2.6090 \qquad (10.118)$$

$$b_2 = \begin{vmatrix} a_0 & a_1 \\ a_3 & a_2 \end{vmatrix} = a_0 a_2 - a_1 a_3 = -1.9536 \qquad (10.119)$$

Since $|b_0|$ is not greater than $|b_2|$, Equation (10.113) has at least one root outside the unit circle.

10.15 STEADY STATE ERROR FOR DISCRETE SYSTEM

For analysis of the steady state behavior of a closed loop discrete control system we consider the block diagram representation shown in Fig. 10.26.

Here, the error can be shown to be $E(z) = \dfrac{R(z)}{1 + G(z)\,H(z)}$

Using the final value theorem (vide Equation 10.17), the steady state error is given by

$$e_{ss} = \lim_{k \to \infty} e(kT) = \lim_{z \to 1} (1 - z^{-1})\, E(z) = \lim_{z \to 1} \frac{(1 - z^{-1})\, R(z)}{1 + G(z)H(z)} \qquad (10.120)$$

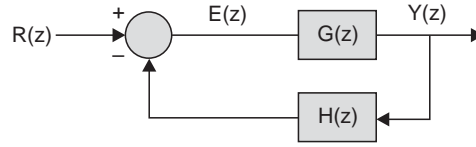



Fig. 10.26 Block diagram of a discrete feedback control system

The steady state error e_{ss} is found to depend not only on the loop gain G(z)H(z) but also on the nature of input R(z), as was the case with analog systems. Proceeding as in the case of analog systems, we can find the following error constants for the discrete systems, for the typical test inputs normally used to assess the steady state performance of a closed loop systems in the laboratory environment.

1. Step input

$$r(t) = Ru_s(t), \quad t \geq 0$$
$$= 0, \quad t < 0 \tag{10.121}$$

The z-transform of $r(t)$ is given by

$$R(z) = \frac{R\,z}{z-1} = \frac{R}{(1-z^{-1})}$$

Substituting this values of R(z) in relation (10.120) we get

$$e_{ss} = \lim_{z \to 1} \frac{(1-z^{-1})\dfrac{R}{(1-z^{-1})}}{1+G(z)H(z)} \lim_{z \to 1} \frac{R}{1+G(z)H(z)}$$

$$= \frac{R}{1+K_p} \text{ when } K_p = \lim_{z \to 1} G(z)H(z) \tag{10.122}$$

$$e_{ss} = 0 \text{ if } K_p = \infty$$

2. Ramp input $\quad r(t) = Rtu_s(t),\ t \geq 0 \tag{10.123}$

$$= 0, \quad t < 0$$

The z-transform of $r(t)$ is given by

$$R(z) = \frac{RTz}{(z-1)^2}$$

Here T is the sampling period. Substituting this value of R(z) in relation (10.120) we get:

$$e_{ss} = \lim_{z \to 1} \frac{(1-z^{-1})\dfrac{RTz}{(z-1)^2}}{1+G(z)H(z)} = \lim_{z \to 1} \frac{RT}{(z-1)[1+G(z)H(z)]}$$

or

$$e_{ss} = \lim_{z \to 1} \frac{R}{\dfrac{1}{T}(z-1)[1+G(z)H(z)]} = \frac{R}{K_v} \tag{10.124}$$

where $K_v = \lim_{z \to 1} \dfrac{(z-1)}{T}[1+G(z)H(z)] = \lim_{z \to 1} \left(\dfrac{1-z^{-1}}{T}\right) G(z)H(z)$ (10.125)

3. Parabolic input

$$r(t) = \begin{cases} \dfrac{Rt^2\, u_s(t)}{2} & , \quad t \ge 0 \\ 0 & , \quad t < 0 \end{cases} \tag{10.126}$$

The z-transform of $r(t)$ is given by

$$R(z) = \frac{RT^2 z(z+1)}{2\,(z-1)^3}$$

Substituting the value of R(z) in relation (10.120), we have

$$e_{ss} = \lim_{z \to 1} (1 - z^{-1}) \frac{\left(\dfrac{RT^2 z(z+1)}{2\,(z-1)^3}\right)}{1 + G(z)H(z)}$$

$$e_{ss} = \lim_{z \to 1} \frac{RT^2\,(z+1)}{2\,(z-1)^2\,[1 + G(z)H(z)]}$$

$$= \lim_{z \to 1} \frac{R}{\dfrac{2\,(z-1)^2}{T^2\,(z+1)}\,[1 + G(z)H(z)]} = \frac{R}{K_a} \tag{10.127}$$

where $K_a = \lim_{z \to 1} \dfrac{2}{z+1}\left(\dfrac{z-1}{T}\right)^2 G(z)H(z) = \lim_{z \to 1}\left(\dfrac{z-1}{T}\right)^2 G(z)H(z) = \lim_{z \to 1}\left(\dfrac{1-z^{-1}}{T}\right)^2 G(z)H(z)$

(10.128)

The error constants K_p, K_v, K_a may be used as design parameters to meet a prescribed e_{ss} (vide Example 10.20 below).

The following points should be remembered while applying the error-constant analysis.

The expressions for step-, ramp-, and parabolic-error constants are applicable in the error analysis only when the input signal is a step function, ramp function, and parabolic function, respectively.

Since the error constants are defined with respect to the loop transfer functions G(z)H(z), strictly, the method is applicable to only the system configuration shown in Figures 10.26. Since the error analysis relies on the use of the final-value theorems of z-transforms, it is important to first check to see if $(1 - z^{-1})E(z)$ has any poles on or outside unity circle in the z-plane.

The steady-state error of a system with an input that is a linear combination of the three basic types of inputs can be determined simply by superimposing the errors due to each input applied separately.

When the system configuration differs from that of Fig. 10.26 we can find out the error signal for the new configuration and apply the final-value theorem following the procedure outlined above.

One of the disadvantages of the error-constant method of analysis is that it does not show the variation of the error with time when the steady-state error is moving towards infinity. Also, the error-constant method is not applicable to sinusoidal inputs, since the final-value theorem cannot be applied to this input with roots on the unity circle.

Example 10.20 In the system of Fig. 10.27, the plant transfer function $G(s) = 1/(s + 1)$ is preceded by a zero order hold circuit. With $D(z)$ as a PI controller with proportionality constant K_p and a sampling period of 0.1 sec, find the limiting values of K_p so that the closed loop system is stable and the steady state error is less than 0.01 to ramp input of unity slope.

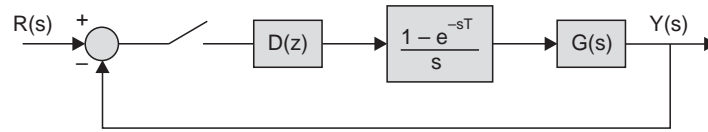



Fig. 10.27 Closed loop system for Example 10.20

Solution: The Proportional Integral (PI) controller in s domain is given by

$$D(s) = \frac{K_I}{s} + K_p$$

The PI controller $D(z)$ in z-domain is found by taking the z-transform of $D(s)$, given by:

$$D(z) = \frac{K_I z}{z-1} + K_p = \frac{(K_I + K_p)\,z - K_p}{z-1}$$

Here the loop gain = $G(z)D(z)$, where $G(z)$ is the Z-transform of $G(s)$ and is given by

$$G(z) = (1 - z^{-1}) \times Z \text{ of } \left[\frac{1}{s(s+1)}\right] = \frac{z-1}{z} \times \left[\frac{z(1-e^{-T})}{(z-1)(z-e^{-T})}\right] = \frac{1-e^{-T}}{z-e^{-T}}$$

Using relation (10.125) for the velocity error constant

$$K_v = \lim_{z \to 1} \left(\frac{1-z^{-1}}{T}\right) [D(z)\, G(z)]$$

$$= \lim_{z \to 1} \left(\frac{1-z^{-1}}{T}\right) \left[\frac{(K_1 + K_p)\,z - K_p}{z-1}\right] \left[\frac{1-e^{-T}}{z-e^{-T}}\right] = \frac{K_I}{T}$$

In order to satisfy the constrain $e_{ss} < 0.01$, we use relation (10.124) to set $\frac{R}{K_v} \le 0.01$; since $R = 1$, $T = 0.1$sec, we get using the value of K_v from above,

$$\frac{T}{K_I} \le 0.01 \quad \Rightarrow \quad K_I \ge 100T = 10$$

The characteristic equation $1 + D(z)\, G(z) = 0$ yields

$$z^2 - [0.953249 - 0.095163\, K_p]z + 0.904837 - 0.095163\, K_p = 0$$

Let us now examine the criteria of stability (10.100) by Jury-Blanchard technique for the above characteristic equation.

Criteria 1 $F(1) > 0$ $\Rightarrow 1 + 0.095163\, K_p - 0.953249 + 0.904837 - 0.095163\, K_p > 0$

$\Rightarrow$ Independent of K_p

Criteria 2 $(-1)^2 F(-1) > 0$ yields

$$1 - 0.095163\, K_p + 0.953249 + 0.904837 - 0.095163\, K_p > 0$$

$\Rightarrow \quad 2.858086 - 0.190326\ K_p > 0$

$\Rightarrow \quad K_p < 15.016792$

Criteria 3 $\quad |\ 0.904837 - 0.095163\ K_p\ | < 1$

$\Rightarrow \quad -1 < K_p < 20.016572$

The limiting values are found from $0.904837 - 0.095163\ K_p = \pm 1$

So considering all the three conditions together we have, $-1 < K_p < 15.016792$ for the steady state error to be less than < 0.01.

Example 10.21 Let us consider the following characteristic equation obtained with a sampling period T = 0.1 sec containing a variable parameter K. We are interested to find the range of K > 0 for which the system is stable.

$$z^3 - 0.1z^2 + 0.2K^z - 0.1\ K = 0 \qquad (10.129)$$

Here
$a_3 = 1$
$a_2 = -0.1$
$a_1 = 0.2\ K$
$a_0 = -0.1\ K$

Applying the conditions in Equation (10.99), we have

$$F(1) = 1 + a_2 + a_1 + a_o$$

$$= 0.9 + 0.1\ K > 0, \qquad (10.130)$$

$$F(-1) = -1 + a_2 - a_1 + a_0$$

$$= -1.1 - 0.3K < 0 \qquad (10.131)$$

$$|\ a_0\ | = |\ 0.1\ K\ | < 1 = a_3 \qquad (10.132)$$

The coefficients in the last row of the Jury tabulation are

$$b_0 = a_0^2 - a_3^2 = 0.01\ K^2 - 1 \qquad (10.133)$$

$$b_1 = a_0a_1 - a_2a_3 = -0.02\ K^2 + 0.1 \qquad (10.134)$$

$$b_2 = a_0a_2 - a_1a_3 = 0.01\ K - 0.2\ K \qquad (10.135)$$

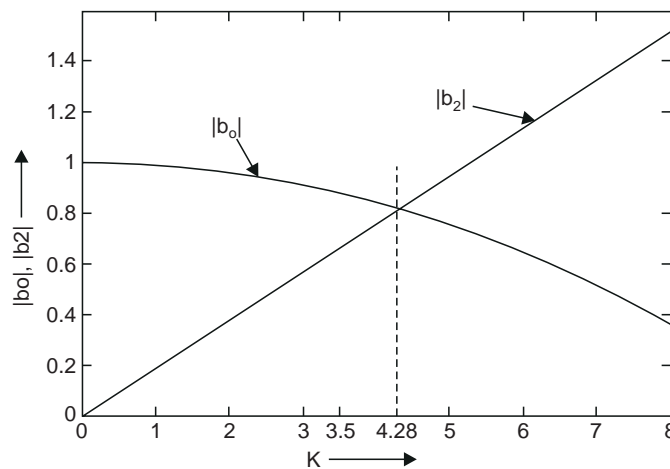



Fig. 10.28 Variation of b_0 and b_2 with K

From Equation (10.130), F(1) > 0 implies $K > 0$ (10.136)

and from (10.132), $| a_0 | < a_3$ yields, $K < 10$ (10.137)

The condition of $| b_0 | > | b_2 |$ leads to

$$| 0.01 K^2 - 1 | > | 0.01 K - 0.2 K | \quad (10.138)$$

which is best solved graphically, as shown in Fig. 10.28.

Thus, the maximum value of K for stability is 0 < K < 4.28, which is a more stringent requirement than that of Equation (10.137). Therefore, the range of K for system-stability when T = 0.1 sec is

$$0 < K < 4.28 \quad (10.139)$$

10.16 STATE FEEDBACK DESIGN FOR DISCRETE SYSTEMS

The design and analysis tools like root loci, Nyquist plot, Bode plot and pole assignment technique presented in connection with continuous systems can be readily adapted to discrete cases. We shall illustrate below the adaptation of all the relations developed for pole assignment design for continuous systems to discrete systems represented in state variable form. For the discrete case, the n-th order plant is described by the state equation:

$$x(k + 1) = A\, x(k) + B\, u(k) \quad (10.140)$$

$$y(k) = C\, x(k) + D\, u(k)$$

and the control law is generated as $u(k) = -K'\, x(k)$ (10.141)

where, $K' = [k_1\ k_2\ .....\ k_n]$ (10.142)

The closed loop equation for the plant is, therefore, obtained from (10.140) as

$$x(k + 1) = (A - BK')\, x(k) \quad (10.143)$$

Choosing the desired pole locations at

$$z_i = \lambda_i, \quad i = 1 \text{ to } n \quad (10.144)$$

the desired characteristic equation becomes

$$\alpha_c(z) = (z - \lambda_1)(z - \lambda_2) \ldots (z - \lambda_n) = z^n + \alpha_{n-1} z^{n-1} + \ldots + \alpha_1 z + \alpha_0 = 0 \quad (10.145)$$

which is made identical, by proper choice of K, with $| zI - A + B K' | = 0$ (10.146)

The state feedback gains are solved by Ackerman's formula, as before,

$$K' = [0\ 0\ 0\ \ldots\ 0\ 1]\ [B\ AB\ \ldots\ldots\ A^{n-2} B\ \ A^{n-1} B]^{-1}\ \alpha_c(A) \quad (10.147)$$

The state feedback coefficients can be found using the MATLAB command

$$K' = \text{acker}(A, B, p), \text{ where } p = [\lambda_1\ \ \lambda_2\ \ \ldots\ \ \lambda_n]$$

10.16.1 Predictor Estimator

The states which are not available for measurement and feedback are to be estimated by digital computer using arrangement similar to that shown in Fig. 9.2 in Chapter 9. However, discrete controllers have two options of implementing the estimators, namely: **Predictor Estimator** and **Current Estimator.** Since, the outputs are measured only at sampling instants and computation of control laws take some finite time, it is convenient, especially for microcomputers, which are not very fast, to generate the control law $u(k)$ for the current instant by using measurements of the output upto $(k - 1)$th instant. Estimators implemented based on this principle are referred to as **Predictor Estimator.** But when the process is slow, the computation time may be negligibly small compared to the sampling period and non-utilization of the current measurement is a wastage of computing resources. Estimators realized with measurements upto and including the current instant is known as **Current Estimator.**

Writing the estimator dynamics as

$$q(k + 1) = F\, q\,(k) + G\, y(k) + H\, u(k) \tag{10.148}$$

and taking its z-transform and following the procedure similar to that for the continuous case, the estimator equation can be written in terms of system matrix as:

$$q(k + 1) = (A - GC)\, q(k) + G\, y(k) + B\, u(k). \tag{10.149}$$

So, the observer in Equation (10.149) is a *predictor observer,* since the estimate at time $(k + 1)T$ is based on the measurement of the output $y(kT)$ at time kT.

Defining the error vector $e(k)$ as $e(k) = x(k) - q(k)$

we can show that the error dynamics is identical with the estimator dynamics.

That is, $e(k + 1) = (A - G_pC)\,[x(k) - q(k)] = [A - G_pC]\, e(k)$

The matrix G_p for the *predictor observer* may be computed by the Ackerman's formula as:

$$G_p = \alpha_c\,(A) \begin{bmatrix} C \\ CA \\ \vdots \\ \vdots \\ CA^{n-1} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 1 \end{bmatrix} \tag{10.150}$$

10.16.2 Current Estimator

We will now consider a full-order current estimator. As before the system model is given by

$$x(k + 1) = Ax(k) + Bu(k) \tag{10.151}$$

$$y(k) = Cx(k)$$

We wish to estimate the state-vector $x(k)$ with the vector $q(k)$. One form of a full-order current observer is given by the two equations

$$\bar{q}\,(k + 1) = A\, q(k) + B\, u(k) \tag{10.152}$$

$$q(k + 1) = \bar{q}\,(k + 1) + G_{co}\, e\,(k + 1) = \bar{q}\,(k + 1) + G_{co}\,[y(k + 1) - C\,\bar{q}\,(k + 1)]$$

In the above equations, $\bar{q}\,(k + 1)$ is the approximate value of estimator state which is initially taken to be equal to the system state at time $(k + 1)T$ (the suffix $_{co}$ is used with the G matrix to indicate the current observer). This approximate value is then corrected in the second equation when the measurement at time $(k + 1)T$ is available. The estimator gain G_{co} determines the weight placed on the difference between measurement at $(k + 1)T$ and the predicted value at that time.

In Equation (10.152) the first estimate $\bar{q}\,(k + 1)$ can be eliminated by substituting the first equation into the second one. The single equation for the estimator is then

$$q(k + 1) = [A - G_{co}CA]\; q(k) + [B - G_{co}\,CB]\; u(k) + G_{co}y\,(k + 1) \tag{10.153}$$

The characteristic equation of the estimator is

$$|\; zI - A + G_{co}CA \;| = 0 \tag{10.154}$$

As in the prediction-observer case, we define the estimation-error vector $e(k)$ by the relationship

$$e(k) = x(k) - q(k) \tag{10.155}$$

Then $e\,(k + 1) = x\,(k + 1) - q(k + 1)$

$= \mathrm{A}x(k) + \mathrm{B}u(k) - [\mathrm{A} - \mathrm{G}_{co}\mathrm{CA}]\, q(k) - [\mathrm{B} - \mathrm{G}_{co}\mathrm{CB}]\, u(k) - \mathrm{G}_{co}\mathrm{C}[\mathrm{A}x\,(k) + \mathrm{B}u(k)]$

$= [\mathrm{A} - \mathrm{G}_{co}\mathrm{CA}]\, [x(k) - q(k)] = [\mathrm{A} - \mathrm{G}_{co}\mathrm{CA}]\, e(k)$

Hence the error vector has dynamics with the same characteristic equation as the estimator (10.153).

Thus for a single measurement, the Ackermann's formula for the current observer is obtained from that for the prediction observer by replacing C with CA . Thus, for the current observer

$$\mathrm{G}_{co} = \alpha_c\,(\mathrm{A}) \begin{bmatrix} \mathrm{CA} \\ \mathrm{CA}^2 \\ \vdots \\ \vdots \\ \vdots \\ \mathrm{CA}^n \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{10.156}$$

The combined plant and full-order current estimator equation may be written, using Equations (10.141), (10.151) and (10.153), as

$$\begin{bmatrix} x(k+1) \\ q(k+1) \end{bmatrix} = \begin{bmatrix} \mathrm{A} & -\mathrm{BK}' \\ \mathrm{G}_{co}\,\mathrm{CA} & \mathrm{A} - \mathrm{BK}' - \mathrm{G}_{co}\mathrm{CA} \end{bmatrix} \begin{bmatrix} x(k) \\ q(k) \end{bmatrix} \tag{10.157}$$

In pole assignment design, placement of poles is a tricky affair. However, placement of dominant poles in s-domain are not that difficult, where we can correlate the damping ratio and time constant rather easily with the time-domain specifications. So, we are interested to find a relation of the damping ratio δ and time constant τ of a second order system in s domain with the z-domain pole parameters.

The roots of the second order characteristic equation $s^2 + 2\delta\omega_n s + \omega_n^2 = 0$ are given by

$$s_{1,2} = -\,\delta\omega_n \pm j\omega_n \sqrt{1-\delta^2}$$

Since, $z = e^{s\mathrm{T}}\,|_{s_{1,2}} = e^{-\,\delta\omega_n \mathrm{T}} \angle \pm \omega_n\,\mathrm{T}\sqrt{1-\delta^2} = r\angle \pm \theta$, where T is the sampling period

(10.158)

We have, $$e^{-\,\delta\omega_n \mathrm{T}} = e^{-\frac{\mathrm{T}}{\tau}} = r \tag{10.159 a}$$

and $$\omega_n \mathrm{T}\sqrt{1-\delta^2} = \theta \tag{10.159b}$$

Hence δ is found to be $$\delta = \frac{-\ln r}{\sqrt{\ln^2 r + \theta^2}} \tag{10.160}$$

and the time constant is given by $$\tau = \frac{1}{\delta\omega_n} = \frac{-\mathrm{T}}{\ln r} \tag{10.161}$$

So, we can find out the equivalent s-domain time constant and damping ratio of a quadratic pole pair in z-domain by using relations (10.161) and (10.160), respectively. Alternatively, if the design specification is given in terms of time constant and damping ratio in s-domain, the corresponding locations of the complex conjugate pole pair in z-domain can be computed from relations (10.158) and (10.159) as $z_1 = re^{j\theta}$ and $z_2 = re^{-j\theta}$, respectively.

Example 10.22 We shall now demonstrate the design of an observer in discrete domain for the problem in Example 9.2 in Chapter 9 for which the system matrices are reproduced below with suffix c to indicate their association with continuous system

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & -4 \end{bmatrix}, b_c = \begin{bmatrix} 0 \\ 100 \end{bmatrix}, c_c = [1\ 0] \text{ and } d_c = [0]$$

Employing the discretization technique of Section 10.11, we obtain the following discrete state equation with sampling period T = 0.05 sec.

$$x(k+1) = \begin{bmatrix} 1 & 0.0453 \\ 0 & 0.8187 \end{bmatrix} x(k) + \begin{bmatrix} 0.1171 \\ 4.5317 \end{bmatrix} u(k)$$

and $$y(k) = [1\ 0]\, x(k)$$

With a control law of $u(k) = -\,x_1(k)$, characteristic equation of the closed loop system becomes $z^2 + 1.7017z + 0.9283 = 0$ with roots at

$$z_1 = 0.8508 + j0.4520,\ z_2 = 0.8508 - j0.4520.$$

Using (10.160) and (10.161) the damping ratio δ and time constant τ are given by

$$\delta = 0.0760,\ \tau = 1.3431$$

Since, the damping ratio of the closed loop poles is too small, the initial condition response is unsatisfactory and it takes about 5.25 sec to settle to the set point. A step disturbance to plant will exhibit overshoot of 81 percent.

Let us decide to increase the damping ratio to a value of $\delta = 0.53$ and time constant to $\tau = 1.0$ for the closed loop system. The closed loop poles in the z-domain are computed using the Equations (10.158) and (10.159) with T = 0.05 sec as

$$z_1 = 0.9482 + j\,0.0760,\ z_2 = 0.9482 - j\,0.0760$$

So the desired characteristic equation is $z^2 - 1.8964z + 0.9048 = 0$

The feedback coefficients are computed from relation (10.147) as

$$K' = [0.0374,\ -0.0181]$$

The closed loop system matrix is found to be,

$$A_f = A - BK' = \begin{bmatrix} 0.9956 & 0.0474 \\ -0.1693 & 0.9007 \end{bmatrix}$$

The step disturbance response shows a over shoot of 14 percent and settling time of 3.1 sec and rise time 0.9 sec. The disturbance to the plant enters as an additional input $B_1 w(k)$ to the state equation (10.140) where $B'_1 = [0\ 10]$.

Example 10.23 Let us now illustrate the design of full-order current observer for the problem in Example 10.22.

We take a time constant for the estimator $\tau = 0.4$ sec which is 2.5 times faster than the smallest system time constant of 1 sec. We also take both the estimator routs to be real and equal. The estimator roots are given by

$$z_{1,2} = e^{-T/\tau} = e^{-0.05/0.4} = 0.8825$$

The observer characteristic polynomial is then

$$\alpha_e(z) = (z - 0.8825)^2 = z^2 - 1.7650z + 0.7786$$

From Equation (10.156) we compute the matrix G_{co} as

$$G_{co} = \begin{bmatrix} 0.04877 \\ 0.10959 \end{bmatrix}$$

From (10.153), the estimator state equations are

$$q(k+1) = [A - G_{co}CA]\, q(k) + [B - G_{co}\, CB]\, u(k) + G_{co}y\,(k+1)$$

$$= \begin{bmatrix} 0.95123 & 0.04310 \\ -0.10954 & 0.81376 \end{bmatrix} q(k) + \begin{bmatrix} 0.04877 \\ 0.10959 \end{bmatrix} y(k+1) + \begin{bmatrix} 0.11136 \\ 4.51890 \end{bmatrix} u(k)$$

But $\qquad u(k) = -K'\, q(k) = 0.0374\, q_1(k) - 0.0181 q_2(k)$

Therefore, the observer dynamics are:

$$q(k+1) = \begin{bmatrix} 0.94706 & 0.045123 \\ -0.2786 & 0.89556 \end{bmatrix} q(k) + \begin{bmatrix} 0.04877 \\ 0.10959 \end{bmatrix} y(k+1)$$

From relation (10.157) the combined plant and full-order current estimator is found as:

$$\begin{bmatrix} x_1(k+1) \\ x_2\,(k+1) \\ q_1\,(k+1) \\ q_2\,(k+1) \end{bmatrix} = \begin{bmatrix} 1 & 0.045317 & -0.0043783 & 0.0021189 \\ 0 & 0.81873 & -0.16949 & 0.082024 \\ 0.048771 & 0.0022102 & 0.94685 & 0.045226 \\ 0.10959 & 0.0049664 & -0.27908 & 0.89579 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2\,(k) \\ q_1\,(k) \\ q_2\,(k) \end{bmatrix}$$

The response $y(k)$ to the initial condition $x'(0) = [1\ 0]$ for full-order current observer-based control system is shown in Fig. 10.29.

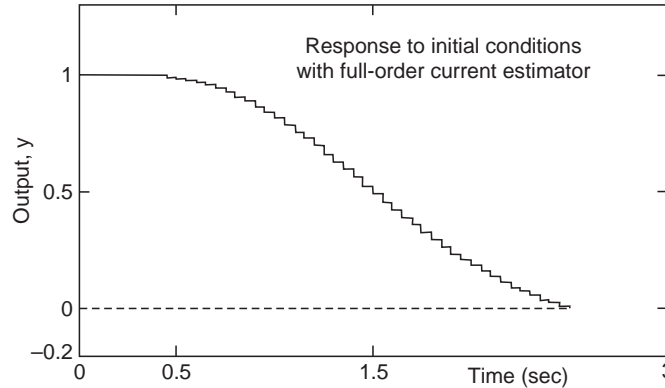



Fig. 10.29 Initial condition response for full-order current observer

10.16.3 Reduced-order Estimator for Discrete Systems

Proceeding in the same way as in the continuous systems we can readily write the reduced-order estimator for digital systems. We denote the measurable states by $x_a(k)$ and the states to be estimated by $x_b(k)$, so that the state vectors are partitioned as:

$$x(k) = \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix}$$

Then the plant state equation of (10.140) can be partitioned as

$$\begin{bmatrix} x_a(k+1) \\ x_b(k+1) \end{bmatrix} = \begin{bmatrix} A_{aa} & A_{ab} \\ A_{ba} & A_{bb} \end{bmatrix} \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix} + \begin{bmatrix} B_a \\ B_b \end{bmatrix} u(k) \qquad (10.162)$$

$$y(k) = [1\ 0] \begin{bmatrix} x_a(k) \\ x_b(k) \end{bmatrix}$$

Following the procedure outlined for continuous systems we can write the equation for the reduced order estimator as

$$q_b\,(k+1) = (A_{bb} - G_r A_{ab})\, q_b\,(k) + G_r\, y(k+1) + (A_{ba} - G_r A_{aa})\, y(k) + (B_b - G_r\, B_a)\, u(k) \qquad (10.163)$$

The observer characteristic equation is

$$\alpha_e\,(z) = |\; zI - A_{bb} + G_r\, A_{ab} \;| = 0 \qquad (10.164)$$

and the estimator gain matrix G_r for the reduced order observer is computed in the same manner from (10.150) used for computing G for full-order observer. For the case of a single measurement [*i.e.*, $y(t)$ is $x_1(t)$] Ackerman's formula is given by

$$G_r = \alpha_c\,(A_{bb}) \begin{bmatrix} A_{ab} \\ A_{ab}\, A_{bb} \\ A_{ab}\, A_{bb}^{\;2} \\ \vdots \\ \vdots \\ A_{ab} A_{bb}^{\;n-2} \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ 0 \\ \vdots \\ \vdots \\ 0 \\ 1 \end{bmatrix} \qquad (10.165)$$

From Equation (10.163), we note that the measurement of $y(k+1)$ is required to estimate $q_b(k+1)$, so the reduced order estimator is a current observer.

The combined plant and reduced order estimator state equations can also be written as

$$\begin{bmatrix} x_a(k+1) \\ q_b(k+1) \end{bmatrix} = \begin{bmatrix} A_{aa} - B_a K'_a & -B_b K'_b \\ G_r CA + A_{ba} C - B_a K'_a C - G_r A_{aa} C & A_{bb} - B_b K'_b - G_r A_{ab} \end{bmatrix} \begin{bmatrix} x_a(k) \\ q_b(k) \end{bmatrix} \qquad (10.166)$$

10.17 PROVISION FOR REFERENCE INPUT

The closed loop system that results when the controller is implemented by state feedback design, is a regulator system, where the design objective is to drive all the states to equilibrium state following an initial condition response or plant disturbance. However, there should be provision for a reference input in order to set the output to a desired value, as was done in continuous system (vide Figures 9.7 and 9.8 of Chapter 9). Another variation to the structure shown in Fig. 9.8, where the estimator is in the feedback path, is shown in Fig. 10.30.

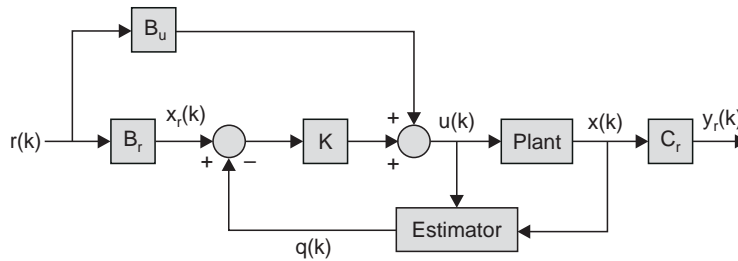



Fig. 10.30 Structure to provide reference input in pole assignment design of discrete system with state estimator

In Fig. 10.30, the input matrices B_r and B_u are chosen to transform the reference input to the desired state vector x_r and desired control $u(k)$. The desired state vector and control in turn are chosen to achieve a desired output y_r, which may be different from the measured output that is used as input to the estimator.

From the Fig. 10.30, if we assume that the matrix B_u is set to zero temporarily, we see that

$$B_r\, r(k) = x_r \quad \text{and } u(k) = -\,K\,[x_r(k) - q(k)] \qquad (10.167)$$

If the plant is at least type 1 system, there will be no steady state error for a step input $r(k)$, such that

$$x(\infty) = x_{ss} = x_r$$

However, if the plant does not contain an integration term, there will be some steady state error and we have to use some control input to maintain x_{ss} to the desired value and hence the output y_r, in turn, to its desired value. The input matrix B_u is chosen to serve this purpose such that

$$u_{ss} = B_u r \qquad (10.168)$$

So for a constant reference input, we can write

$$B_r\, r = x_r = x_{ss} \qquad (10.169)$$

$$C_r\, x_{ss} = y_r = r \qquad (10.170)$$

From (10.169) and (10.170) we have,

$$C_r\, B_r\, r = r, \quad \text{or } C_r\, B_r = I \qquad (10.171)$$

At steady state, the system equation (10.140) can be written as

$$x_{ss} = A\, x_{ss} + Bu_{ss} \quad \text{or} \quad (A - I)\, x_{ss} + Bu_{ss} = 0$$

Using the values of x_{ss} from (10.169) and u_{ss} from (10.168), we have

$$(A - I)\, B_r\, r + B\, B_u\, r = 0$$

or

$$(A - I)\, B_r + B\, B_u = 0 \qquad (10.172)$$

The Equations (10.171) and (10.172) can be written in matrix form as:

$$\begin{bmatrix} A-I & B \\ C_r & 0 \end{bmatrix} \begin{bmatrix} B_r \\ B_u \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix}$$

so we get the values of the input matrices as:

$$\begin{bmatrix} B_r \\ B_u \end{bmatrix} = \begin{bmatrix} A-I & B \\ C_r & 0 \end{bmatrix}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \qquad (10.173)$$

The structure of Fig. 10.30 is similar to that of Fig. 9.8 as can be verified by replacing the block B_r in Fig. 9.8 with $(B_u + KB_r)$.

MATLAB SCRIPTS

```
% Script_Fig10.15a
% Amplitude plot of Zero Order Hold
y3=[ ]
x=0:pi/20:4.5*pi; L=length(x);
for i=1:L
y1=x(i) ;
y2=abs(sin(y1)/y1);                % refer to equation (10.56)
y3=[y3,y2]
end
plot(x/pi,y3)
xlabel('\omega/omega s'), ylabel('|Gho|/T')
```

```
% Script_Fig10.19a
% Amplitude plot of First Order Hold
y5=[ ]
x=0:pi/20:4.5*pi;
L=length(x);
for i=1:L
y1=x(i)
y2=(sin(y1)/y1);                                    % refer to equation (10.59b)
y3=power(y2,2)
x1=1+4*power(y1,2)
x2=power(x1,0.5)
y4=x2*y3
y5=[y5,y4]
end
plot(x/pi,y5)
xlabel('\omega/\omega s \rightarrow'), ylabel('|Gh1|/T \rightarrow')
```

```
% Script_Fig10.19b
% Phase plot of First order hold
y3=[ ]
x=0:pi/20:4.5*pi;
L=length(x);
for i=1:L
y1=x(i)
%y1=pi*x
y2=(180/pi)*(atan(2*y1) – 2*y1)                     %Ref Equation (10.59.c)
y3=[y3,y2]
end
plot(xpi, y3)
 xlabel('\omega/\omega s'), ylabel('Phase angle of Gh1')
```

```
% Script_Fig10.28
% Plot of Fig. 10.28
syms a0 a1 a2 a3 b0 b1 b2 K
bbo=[ ];
bb2=[ ];
a0=–0.1*K; a1=0.2*K;a2=–0.1;a3=1.0;
K=0.0:0.025:8;
bo=abs(0.01*(power(K,2))–1);
b2= abs(-19/100*K);
bbo=[bbo, bo];
bb2=[bb2,b2];
plot(K, bbo), grid
hold on
plot(K,bb2)
xlabel('K \rightarrow'), ylabel('bo, b2\rightarrow')
```

```
% Script_Fig10.29
%Design of a full order current observer
%and its initial condition response
a=[0 1;0 –4];b=[0; 100];c=[1 0];d=[0];
T=0.05;sys=ss(a, b, c, d);sys1=c2d(sys, T);
[a1,b1,c1,d1]=ssdata(sys1);
format short; tou=0.4; z1=exp(-T/tou);
p1=[z1 conj(z1)]; p2=poly(p1);
Gc=acker(a1',a1'*c1',p1)'
afgc=a1-Gc*c1*a1; gca=Gc*c1*a1
k=[0.0374 –0.0181]; bk=b1*k
a_obs=afgc-bk; p_obs=[a1 –bk; gca a_obs]
x10=[1 0 0 0]; b2=[0 ;1 ;0 ;1];c2=[1 0 0 0];
sys3=ss(p_obs,b2,c2,d1,T);
initial(sys3,x10);
 xlabel('Time'), ylabel('output, y')
```

REVIEW EXERCISE

RE10.1 Find the z-transforms of F(s) given below:

(*a*) $F(s) = \dfrac{4}{s(s^2+4)}$; **Ans:** $F(z) = \dfrac{z}{z-1} - \dfrac{z(z-\cos 2T)}{z^2 - 2z\cos 2T + 1}$, T = sampling period

(*b*) $F(s) = \dfrac{4}{s^2(s^2+4)}$, **Ans:** $F(z) = \dfrac{Tz}{(z-1)^2} - \dfrac{1}{2}\left(\dfrac{z\sin 2T}{z^2 - 2z\cos 2T + 1}\right)$

(*c*) $F(s) = \dfrac{2(s+1)}{s(s+5)}$, **Ans:** $F(z) = \dfrac{2z}{5(z-1)} - \dfrac{8}{5}\left(\dfrac{z}{z-e^{-5T}}\right)$

RE10.2 Find the value of $y(kT)$ for $k = 0, 1, 2 \ldots$ for the system $\dfrac{Y(z)}{R(z)} = \dfrac{z-0.2}{4(z-0.8)}$ with T = 1 sec and $r(t) = 1,\ t \geq 0$. Also find the value of $y(kT)$ as $k \to \infty$ **.**

RE10.3 Find the range of K > 0 for which the closed loop systems with loop gain $G(z)H(z)$ given below are stable.

(*a*) $\dfrac{K(2z^2 - 0.5z - 0.4)}{z^3 - 0.75z^2 + 0.25z - 0.15}$, T = 0.1 sec, **Ans:** $K \leq 1.02$

(*b*) $\dfrac{K(z^2 + 2z - 0.75)}{z^3 - 1.6z^2 + 0.65z - 0.05}$, T = 1 sec , **Ans:** $K \leq 0.31$

(*c*) $\dfrac{K(z^2 - 1.5z + 0.5)}{z^3 - 1.4z^2 + z - 0.2}$, T = 0.1 sec, **Ans:** $K \leq 1.2$

(*d*) $\dfrac{K(0.5z - 0.2)}{z^2 - 1.2z + 0.4}$, T = 0.1 sec, **Ans:** $K \leq 3.7143$

(*e*) $\dfrac{K(z - 0.25)}{z^2 - 0.5z + 0.125}$, T = 0.05 sec, **Ans:** $K \leq 1.3$

(*f*) $\dfrac{K(z^2 - 1.5z + 0.6)}{z^3 - 1.8z^2 + 1.5z - 0.4}$, T = 0.02 sec. **Ans:** $K \le 1.52$

RE10.4 For the problem in RE10.3 (*e*) above if H(*z*) = 1 and K = 0.4, find the per cent overshoot and settling time for the system when the loop is closed and it is subjected to a unity step input.
Ans: Over shoot = 23.3%, t_s = 0.15 sec

RE10.5 The transfer function of a Satellite Attitude Control system is given by: $G(s) = \dfrac{1}{s^2}$

Obtain the discrete state variable representation of the satellite attitude-control system with a sample period of T = 0.1 sec. Design a control law for the system to place poles of the closed-loop characteristic equation in the *z*-plane corresponding to the s-plane poles at $s = -2 \pm j4$ rad / sec.

Solution: We found from the review exercise RE9.2, that the system matrices for continuous model are given by:

$$A_c = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \; B_c = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \; C_c = [1 \; 0], \; d_c = 0$$

With a sample period of T = 0.1 sec, the discrete system matrices A and B are found by using relations (10.65) and, (10.68) and are respectively, given by:

$$A = \begin{bmatrix} 1 & 0.1 \\ 0 & 1 \end{bmatrix}, \; B = \begin{bmatrix} 0.0050 \\ 0.1000 \end{bmatrix} \text{ and } C = C_c = [1 \; 0], \; d = d_c = 0$$

The damping ratio and time constants corresponding to the *s*-domain poles at $s = -2 \pm j4$ are, respectively, $\delta = 0.4472$ and $\tau = 0.5000$. With T = 0.1 sec, and using relations (10.159 *a*) and (10.159 *b*) we find $r = 0.8187$ and $\theta = 0.4000$ rad. Therefore, *z*-domain poles are given by z = [0.7541 + 0.3188*i* 0.7541 – 0.3188*i*] and the desired characteristic polynomial is found as: $\alpha_d(z) = z^2 - 1.5082\,z + 0.670$, so the feedback control coefficients are found from (10.147) as K′ = [16.2128 4.1074].

The same results can be verified with the following MATLAB script.

```
A1 = [0 1; 0 0]; b1 = [0; 1]; c1 =[1 0]; d1 = 0;
T = 0.1;
sys=ss(A1,b1,c1,d1);
sys1=c2d(sys,0.1);
[A, b, c, d]=ssdata(sys1);
s1s2=[–2+4i –2-4i];            % given poles in the s-domain
p=poly(s1s2);                  % characteristic equation in s-domain, s^2+2*delta*wn*s+wn^2
wn=power(p(3),0.5);            % p(1) = 1, p(2) = 2*delta*wn and p(3) = wn^2
tou=2/p(2);
delta=1/(tou*wn);
r=exp(–T/tou);
d1=1-power(delta,2);
d2=power(d1,0.5)/delta;
theta=d2*(–log(r));
p1=[r*exp(j*theta) r*exp(–j*theta)];
K=acker(A,b,p1)
```

This yields K′ = [16.2128 4.1074].

Or still better

```
A1=[ 0 1; 0 0]; b1=[0;1];c1=[1 0]; d1=0;
T= 0.1;
sys=ss(A1,b1,c1,d1);
```

```
sys1=c2d(sys,0.1);
[A,b,c,d] = ssdata(sys1);
g=zpk([],[–2+4i –2–4i],1);        % desired s-domain transfer function
g1=c2d(g, T);                     % equivalent z-domain transfer function
p1=eig(g1)                        % equivalent z-domain roots
K=acker(A,b,p1)
```

PROBLEMS

P10.1 For the open-loop system: $G(s) = \dfrac{Y(s)}{U(s)} = \dfrac{1}{s^2 + 0.25\,s + 1}$

(*a*) with a sampling period of T = 0.1 sec, find the discrete state variable representation.

(*b*) Find the full state feedback that provides equivalent *s*-plane poles at $s = -\,1.5000 \pm j\,2.0000$

(*c*) Also find the closed-loop system response to an initial value of $y'(0) = [1\ 0]$

P10.2 For the open-loop system: $G(s) = \dfrac{Y(s)}{U(s)} = \dfrac{1}{s^2\,(s + 2.5)}$,

(*a*) find the discrete state variable representation with ZOH and a sampling period of T = 0.01 sec.

(*b*) Find the state feedback coefficients **K** such that the settling time $t_s < 1$ sec

(*c*) Find the closed loop system response to an initial value of $y'(0) = [1\ 0]$ and check if specification on t_s is satisfied.

P10.3 The open-loop transfer function G(*s*) given below is preceded by a ZOH with a sampling rate of 0.04 sec.

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^2}$$

(*a*) Find the feedback gain **K** so that the closed loop poles are at $s = -\,5.000 \pm j5.000$. Find the estimator gain G so that the estimator poles have an equivalent *s*-plane pole at $s = -\,10.000 \pm j10.000$

(*b*) Determine the discrete transfer function of the compensation.

(*c*) Design a lead compensation using the procedure of Chapter 8, so that the equivalent *s*-plane natural frequency $w_n \cong 10$ rad/sec and $\delta = 0.7$.

(*d*) Compare the compensation transfer functions from (*b*) and (*c*) and discuss the differences.

P10.4 For the system in Problem P10.3, design the controller and estimator so that the closed-loop unit step response to a command input has a rise time $t_r < 150$ msec and an overshoot $M_p < 12\%$ using the state command structure of Fig. 10.30. Also, plot the step response and check that the specifications are met.

P10.5 For the open-loop system

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{s^2(s^2 + 400)}$$

design the controller and estimator so that the closed-loop unit step response to a command input has a rise time $t_r < 250$ msec and an overshoot $M_p < 12\%$ when using the state command structure of Fig. 10.30. Check that the specifications are met by plotting the step response.

P10.6 For the magnetic levitation system described by the equations:

$$\ddot{x} = 900\,x + 15\,u$$

(*a*) select a sampling period of 0.01 sec and design a state feedback controller to meet the specifications that settling time is less than 0.4 sec and overshoot to an initial offset in *x* is less than 15%.

(*b*) Design a reduced-order estimator for $\dot{x}$ for this system such that the error-settling time will be less than 0.1 sec.

(*c*) Plot step responses of x, $\dot{x}$ and u for an initial displacement $x(0)' = [1\ 0]$.

(*d*) Introduce a command reference as in Fig. 10.30 and plot the step response.

P10.7 For the open-loop system

$$G(z) = \frac{y(z)}{u(z)} = \frac{y(z)}{W(z)} = \frac{0.1200(z + 0.9500)}{z^2 - 1.6718z + 0.9048}$$

(*a*) Find the feedback coefficients and estimator gain G that will place closed loop poles at $z = 0.6 \pm j0.2$ and estimator poles at $q = 0.4 \pm j0.3$.

(*b*) Plot the response of $y(k)$ for a unit step input using the state command structure of Fig. 10.30. What would be the value of the steady-state error if $B_u = 0$.

(*c*) Determine the steady state value of $y(k)$ in presence of input disturbance, W.

CHAPTER

11

Optimal Control

11.1 INTRODUCTION

The classical design methods use performance specifications like Phase Margin, Gain Margin, bandwidth, overshoot and damping ratio which will lead to systems that will meet the prescribed specifications, but are not optimal in time or in fuel consumption (control energy). There are many control situations where we are concerned with the design of a controller to optimize a given performance index.

Since the early fifties the issues concerning dynamic optimization have received a lot of attention within the framework of control theory. The optimal control problem usually concerns the following three components:

(*a*) *The equations* which constitute the model of the controlled system;

(*b*) The criterion, referred to as *the performance index,* according to which the system behavior has to be evaluated;

(*c*) The *set of constraints* active on the system state, output, control variables, that are yet to be accounted for by the system model.

Preparation of a mathematical model is a general control problem and is not specific to the optimization problem. In many systems, we must incorporate some method of identification of system parameters which may vary as a function of the environment. Sometimes the plant is well defined, but the states are not available, which must be estimated in presence of plant disturbance and measurement noise.

The choice of performance index depends on the objective to be realized by the control system. It may be related to (*a*) the minimization of time (*b*) regulation of a state or output over extended period of time (*c*) minimization of fuel consumption (*d*) minimization of energy (*e*) maximization of productivity and so on.

The constraints may arise due to the physical nature of the components and operating conditions : the thrust from a booster rocket is finite and limited, the torque of motor is finite, the output of an amplifier is limited by saturation, the speed of a vehicle is finite and so on .

A fairly precise idea of the kind of optimal control problems to be discussed here is given by the following simple examples.

11.2 OPTIMAL CONTROL PROBLEM

1. A frequently encountered problem in space application is the so-called attitude control problem, which in its simplest formulation, consists in orienting an object (satellite) in a specified direction starting from an arbitrary orientation. Letting t_i and t_f denote the extreme values of the control interval, a significant feature of the control problem is the presence of an integral of the form

$$\int_{t_i}^{t_f} f(u(t))dt \le f^{\,0} \tag{11.1}$$

which emphasize the need that the required maneuver be accomplished by consuming a quantity of fuel not exceeding a given bound; the function f quantifies the instantaneous consumption which is assumed to depend on the value of the control variable u applied at that instant.

2. The requirement of keeping the state of a system as close as possible to x_d after a disturbance has occurred, gives rise to an important class of optimal control problems. The perturbation δu to be given to the control u_d has to be determined in such a way as to make $\delta x = x - x_d$ small. Here u_d is the control, which generates x_d in unperturbed condition. In general, it is also desireable that the δu be small hence these requirements can be taken into account by looking for the minimization of the performance index.

$$J = \int_{t_i}^{t_f} \left[\sum_{j=1}^{n} q_j\ \delta x_j^2\ (t) + \sum_{j=1}^{m} r_j\ \delta u_j^2(t) \right] dt,\ q_j > 0,\ r_j > 0 \tag{11.2}$$

In this context the initial time t_i (when $\delta x \neq 0$ because of previous disturbance) and final time t_f are given. The initial state $\delta x(t_i)$ is also given, even if generic, while the final state has to be considered as free.

A special yet particularly interesting case is the one where x_d is an equilibrium state (origin). If the control interval is unbounded ($t_f \to \infty$), the problem can be seen as a stabilization problem since, the minimization of the performance index above ensures, whenever feasible, the system states to approach the origin asymptotically.

3. A typical positioning problem consists in transferring the state x of the control system from a given initial point P_i to the neighborhood of the point P_f with coordinates x_j. The transfer has to be accomplished in short time requiring control actions u of limited magnitude. The problem is characterized by a given initial state variable x_i and time t_i, when the performance index is given in the form :

$$J = \int_{t_i}^{t_f} \sum_{j=1}^{m} r_j u_j^2(t)\ dt + t_f + \sum_{j=1}^{n} s_j\ [x_j(t_f) - x_j]^2, \qquad r_j > 0,\ s_j > 0 \tag{11.3}$$

In general, the problem formulation may also include a set of constraints which set bounds on some of the state variables, typically velocities and/or positions; they can be expressed by means of the relations of the type $f(x(t), t) \le 0$ to be satisfied along the whole control interval.

Example 11.1 One of the less complicated statements of the so-called rendezvous problem makes reference to a point object which is the target; its position is specified by a vector $x_b(t)$ of known functions of time. A second point object must be driven, by suitably selecting a vector u of time functions (the control variables), so as to meet the target (see Fig. 11.1). By denoting the position of the second object at time t with $q(t)$, the *rendezvous* problem amounts to satisfying the relation $x(\tau) = x_b(t)$ at some time, τ, where $x = [q\dot{q}]'$ and $x_b = [q_b \dot{q}_b]'$.

Starting from this basic requirement a series of different scenarios may be encountered.

Case 1: The initial state x_i and the initial time t_i are fixed but the final time t_f, when the *rendezvous* takes place, is free and the performance index J given below is to be minimized.

$$J = \int_{t_i}^{t_f} dt \tag{11.4}$$

Besides, the constraint $x(t_f) = x_b(t_f)$ (which is peculiar to the case), other requirements can be set forth, as for instance, $u_m \leq u(t) \leq u_M$ which account for limits on the control actions.

Case 2: The initial state x_i, the initial time t_i and the final time t_f are given. The final state is only partially specified (for instance, the final position is given, while the final velocity is free inside a certain set of values) and the performance index aims at evaluating the global control effort (to be minimized) by means of an expression of the kind

$$J = \int_{t_i}^{t_f} \sum_{j=1}^{m} r_j u_j^2(t)dt, r_j > 0$$

where u_i is the ith component of the control variable u. The peculiar constraint of the case is $x(\tau) = x_b(\tau)$, where the time τ when the *rendezvous* takes place must satisfy the condition $t_i < \tau < t_f$ and may or may not be specified.

Case 3: This particular version of the rendezvous problem is sometimes referred to as the interception problem. The initial state x_i may or may not be completely specified, while both the initial time t_i and final time t_f are to be selected under the obvious constrain $t_i < t_f \leq T$. The final state is free and the performance index is as in case 2. The peculiar constraint of the case involves some of the state variables only (the positions), precisely, $q(\tau) = q_b(\tau)$, where the time τ when interception takes place may or may not be given and satisfies the condition $t_i < \tau < t_f$.

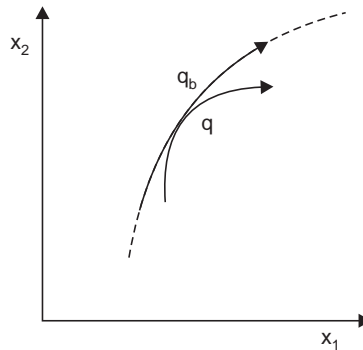



Fig. 11.1 The rendezvous problem

These examples suggest the structure of the optimal control problems and are characterized by:

(*a*) *The equations, which describe the dynamic behavior of the system.* They are either differential or algebraic and their number is finite.

(*b*) *The set of allowable initial states:* It provides the complete or partial specification of the initial state as well as its freedom.

(*c*) *The set of allowable final states:* It provides the complete or partial specification of the final state as well as its freedom.

(*d*) *The performance index:* It is constituted by two terms, which need not both be present: one of them is the integral of a function of the state, control and time, whereas the second one is a function of the final state and time.

(*e*) *A set of constraints on the state and/or control variables:* They can account for integral or instantaneous requirements and involve both equality and inequality relations.

(*f*) *The control interval:* It can be pre-assigned or free, of finite or infinite duration.

The optimization problem is a very important issue in automation, which is concerned with the development of systems that is intended to operate without direct human intervention. In general, the design of optimum control is considered as a variational problem. There are number of variational methods for maximizing or minimizing a functional over a function space (vide section 11.4). Among these, the most widely used methods in the design of control systems are:

(*a*) The calculus of variations

(*b*) The minimum principle

(*c*) The dynamic programming

In all the above cases the objective is to find the optimum control sequence or control law in such a way that the given functional of the performance indices is maximized or minimized. It is interesting to note that the use of variational principles is common in all three methods. Each of these methods, in turn, is related to well known techniques in classical mechanics: the first to the Euler Lagrange equation, the second to the Hamiltonian principle and the third to the Hamilton Jacobi theory. The minimum principle of Pontryagin [33] employs the direct procedure of the calculus of variations, whereas the dynamic programming, developed by Bellman [34], apart from following the variational principle, employs recurrence relationship or the algorithm of partial differential equations.

In this Chapter, we shall mainly confine ourselves to the discussions of calculus of variations and the minimum principle.

11.3 PERFORMANCE INDEX

The performance of a controller is assessed with reference to a mathematical function known as performance index. In modern control theory the most popular choice of the mathematical function is the *integral performance index* of the general form :

$$J = \int_{t_i}^{t_f} L(x, u, t)\, dt \tag{11.5}$$

The performance of the system is said to be optimal over the interval t_i to t_f if the value of the performance index in (11.5) is optimal (minimum or maximum depending on the situation). It is to be noted that mathematical functions other than the *integral performance index* may also be considered to assess the performance of a system. However, we confine our attention to it since it is adequate to serve our purpose.

11.4 CALCULUS OF VARIATIONS

Finding the maximum (or minimum) of a function x is a commonly occurring problem in engineering, where the argument of the function x is an independent variable like time. With reference to the performance index in relation (11.5), we see that x itself is an argument of another function L. In such a case, the function L is called a *functional*—a function whose independent variables are functions rather than a number. So one has to deal with the issue of finding the optimal values of a functional in connection with the design of an optimal controller, together with finding the value of the function(s) that makes the functional optimal.

In order to tackle the problem of finding the optimal values of functionals, we have to take the help of variational calculus. The concepts and methods of variational calculus are very similar to those employed for computing maxima and minima of a function by the theory

of differential calculus. So, it will be of help to review briefly the theory of ordinary maxima and minima and extend it to the case of variational calculus. In the section below, we briefly review the theory of ordinary maxima and minima of differential calculus followed by a discussion on the elements of variation calculus and some related mathematical concepts.

11.4.1 Functions and Functionals

Let us begin with a formal definition of a function.

Definition 11.1 A *function* x is a rule of correspondence (mapping) that assigns to each element t in a certain set D, called the *domain*, a unique element in a set R, called the *range*.

That is, if t and x be two real variables so related that corresponding to every value of t within a defined domain we get a definite value of x, then x is said to be a function of t defined in this domain.

Example 11.2 Let z_1, z_2 , ... z_n be the coordinates of a point z in n-dimensional Euclidean space and

$$f(z) = \sqrt{z_1^2 + z_2^2 \ldots + z_n^2} \tag{11.6}$$

then f is a function of the coordinates of the point z and the real number assigned by f to a given coordinate value is the distance of the point z from the origin.

One should carefully note the difference between a function f and the value of the function $f(z)$ at a point z. Similar difference exists between a functional and its values for specific arguments.

The definition of a functional parallels that of a function.

Definition 11.2 A *functional* L is a rule of correspondence (mapping) that assigns to each function $\mathbf{x}$ in a certain class Ω, called the *domain,* a unique real number in a set R, called the *range*.

It is to be noted that the domain of a functional is a class of functions and intuitively, we might say, that a functional is a "function of a function".

Definition 11.3 L is a *linear functional* of x if and only if it satisfies the *principle of homogeneity*

$$L(kx) = kL(x) \tag{11.7}$$

for all $x \in \Omega$ and for all real numbers k such that $kx \in \Omega$, and the *principle of additivity (superposition)*

$$L(x_1 + x_2) = L(x_1) + L(x_2) \tag{11.8}$$

for all x_1, x_2, and $(x_1 + x_2)$ in Ω .

Example 11.3 Consider the functional

$$L(x) = \int_{t_i}^{t_f} x(t)\, dt \tag{11.9}$$

where x is a continuous function of t. Let us see if this functional satisfies the principles of homogeneity and additivity.

Now $$kL(x) = k \int_{t_i}^{t_f} x(t)\, dt \tag{11.10a}$$

and $$L(kx) = \int_{t_i}^{t_f} kx(t)\, dt = k \int_{t_i}^{t_f} x(t)\, dt \tag{11.10b}$$

therefore, $$L(kx) = kL(x) \quad (11.10c)$$

for all real k and for all x and kx in Ω.

Again,

$$L(x_1 + x_2) = \int_{t_i}^{t_f} [x_1(t) + x_2(t)]\, dt = \int_{t_i}^{t_f} x_1(t)\, dt + \int_{t_i}^{t_f} x_2(t)\, dt \quad (11.11a)$$

$$L(x_1) = \int_{t_i}^{t_f} x_1(t)\, dt \quad (11.11b)$$

and $$L(x_2) = \int_{t_i}^{t_f} x_2(t)\, dt \quad (11.11c)$$

therefore, $$L(x_1 + x_2) = L(x_1) + L(x_2) \quad (11.11d)$$

for all x_1, x_2, and $x_1 + x_2$ in Ω .

The functional in (11.9) is linear, since both homogeneity *and* additivity are satisfied.

A. Closeness of Functions

If two points are said to be close to one another, a geometric interpretation immediately comes to mind. But what is implied when two *functions* are said to be close to each other? In order to give "closeness" a quantitative measure, the concept of a norm is introduced.

Definition 11.4 The p norm of a point z, in n-dimensional Euclidean space is denoted by $\| z \|_p$ and is defined as :

$$\| z \|_p = [\,| z_1^p | + | z_2^p | + ... + | z_n^p |\,]^{1/p},\ p \in [1\ \infty) \quad (11.12)$$

It satisfies the following properties:

(*a*) $\| z \|_p \geq 0$ and $\| z \|_p = 0$ if and only if $z = 0$. (11.13*a*)

(*b*) $\| kz \|_p = | k | \,.\, \| z \|_p$ for all real numbers k (11.13*b*)

(*c*) $\| z_1 + z_2 \|_p \leq \| z_1 \|_p + \| z_2 \|_p$ (11.13*c*)

Two points $\mathbf{z}_1$ and $\mathbf{z}_2$ are said to be close to each other, if the norm $\| z_1 - z_2 \|_p$ is small.

It is to be noted that with $p = 2$ the norm $\| z \|_2$ gives the Euclidean distance of the z vector from the origin, and with $p = 1$, the norm $\| z \|_1$ gives the Manhattan or Taxicab distance of the vector from the origin. Unless otherwise specified, the symbol $\| z \|$ will stand for Euclidean norm.

Denoting the coordinates of a point $\mathbf{z}_i$ by $\mathbf{z}_{i1}$ and $\mathbf{z}_{i2}$ in a two dimensional space ($n = 2$), the distance of the point $\mathbf{z}_2$ from the point $\mathbf{z}_1$ measured with Euclidean norm, ($p = 2$ in 11.12), is given by

$$\| \mathbf{z}_1 - \mathbf{z}_2 \|_2 = \sqrt{(z_{11} - z_{21})^2 + (z_{12} - z_{22})^2}$$

Therefore, the distance of the point $\mathbf{z}_2$ from the point $\mathbf{z}_1$ is less than a small number δ, written as $\| \mathbf{z}_1 - \mathbf{z}_2 \|_2 < \delta$, implies that the point $\mathbf{z}_2$ is located within a radius of δ measured from $\mathbf{z}_1$ using the Euclidean norm (see Fig. 11.2 *a*). On the other hand, the taxicab distance between the points $\mathbf{z}_1$ and $\mathbf{z}_2$, when measured with $p = 1$ norm, is given by

$$\| \mathbf{z}_1 - \mathbf{z}_2 \|_1 = | z_{11} - z_{21} | + | z_{12} - z_{22} |$$

So, the distance of the point $\mathbf{z}_2$ from the point $\mathbf{z}_1$ is less than a small number δ, written as $\| z_1 - z_2 \|_1 < \delta$, implies that the point $\mathbf{z}_2$ must lie within a rectangle with diagonals of length $2\,\delta$, $\mathbf{z}_1$ lying at the meeting point of the diagonals (see Fig. 11.2 *b*).

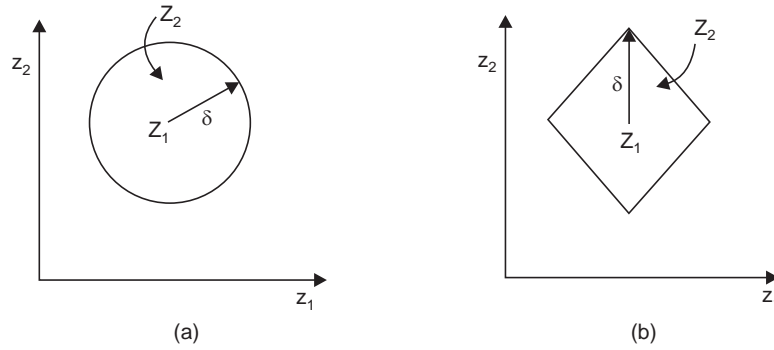



Fig. 11.2 (a) The set of points satisfying the Euclidean norm $\| z_1 - z_2 \|_2 < \delta$; (b) The set of points satisfying the norm $\| z_1 - z_2 \|_1 < \delta$.

Definition 11.5 The norm of a function $x \in \Omega$, defined for $\mathbf{t} \in [\mathbf{t_i}, \mathbf{t_f}]$, is denoted by $\| x \|$, and it satisfies the following properties :

(a) $\| x \| \geq 0$ and $\| x \| = 0$ if and only if $x(t) = 0$ for all $t \in [t_i, t_f]$ (11.14a)

(b) $\| kx \| = | k | . \| x \|$ for all real numbers k . (11.14b)

(c) $\| x_1 + x_2 \| \leq \| x_1 \| + \| x_2 \|$. (11.14c)

In order to compare the closeness of two functions $\mathbf{y_1}$ and $\mathbf{y_2}$ that are defined for $t \in [t_i, t_f]$, we take norm of their difference as $\| e(t) \| = \| y_1(t) - y_2(t) \|$

Intuitively, the norm of the difference of two functions y_1 and y_2 should be zero if the functions are identical ($y_1 = y_2$), small if the functions are "close", and large if the functions are "far apart".

B. Increment of a Functional

In order to find the extremum (maximum or minimum) value of a function, we introduce the concept of increment of a function.

If the function f is defined for the elements of z and $z + \Delta z$, then the *increment* of f, denoted by Δf, is found as :

$$\Delta f(z, \Delta z) = f(z + \Delta z) - f(z) \tag{11.15}$$

In general, Δf depends both on z and Δz.

The increment of a functional is defined in an analogous manner.

Definition 11.6 If x and $x + \delta x$ are functions for which the functional L is defined, then the *increment of* L, denoted by ΔL, is defined as :

$$\Delta L(x, \delta x) = L(x + \delta x) - L(x)$$

The change in the function δx is called the *variation* of the function x.

C. The Variation of a Functional

After we have defined the increments of functions and functionals, we would like to take up the concept of variation of a functional. The *variation of functionals* plays the same role in determining its extreme values as the differential does in finding maxima and minima of *functions.* So, let us first define the differential of a function.

Definition 11.7 The increment of a function of n variables can be written as

$$\Delta f(z, \Delta z) = df(z, \Delta z) + g(z, \Delta z) . \| \Delta z \| \tag{11.16}$$

where df is a linear function of Δz.

If $\lim_{\|\Delta z\| \to 0} \{g(z, \Delta z)\} = 0$, then f is said to be *differentiable* at z, and df is the *differential of* f at the point z.

If f is a differentiable function of one variable t, then the differential can be written

$$df(t, \Delta t) = f'(t)\,\Delta t \tag{11.17}$$

$f'(t)$ is called the *derivative* of f at t. Fig. 11.3 shows a geometric interpretation of the increment Δf, the differential df, and the derivative $f'(t)$. The derivative $f'(t_1)$ is the slope of the line that is tangent to f at the time t_1 and $f'(t)\Delta t$ is a first-order (linear) approximation to Δf such that the smaller the increment Δt, the better is the approximation.

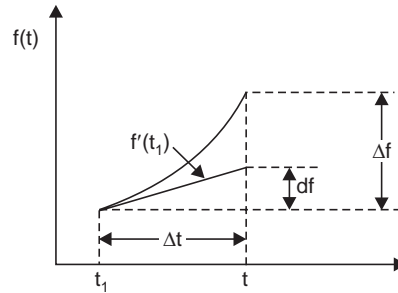



Fig. 11.3 Geometric interpretation of increment Δf of $f(t)$ in terms of differential df and slope $f'(t_1)$ at point t_1

The definition 11.7 can be used to develop a rule for finding the differential of a function. So, if f is a differentiable function of n variables, the differential df is given by

$$df = \frac{\partial f}{\partial z_1}\Delta z_1 + \frac{\partial f}{\partial z_2}\Delta z_2 + \ldots.. + \frac{\partial f}{\partial z_n}\Delta z_n \tag{11.18}$$

We shall now develop a formal procedure for finding the *variation of a functional* instead of starting each time from the definition.

Definition 11.8 Now the increment of a functional can be written as

$$\Delta L(x, \delta x) = \delta L(x, \delta x) + g(x, \delta x).\,\|\delta x\|\text{ ; where } \delta L \text{ is linear in } \delta x.$$

If, $\lim_{\|\delta x\| \to 0} \{g(x, \delta x)\} = 0$, then L is said to be *differentiable* on x and δL is the *variation of* L evaluated for the function x.

Example 11.4 Find the variation of the functional given below, where x is a continuous scalar function defined in the interval $t \in [0, 1]$.

$$J(x) = \int_0^1 [x^2(t) + 4x(t)]\,dt \tag{11.19}$$

The increment of J is given by

$$\Delta J(x, \delta x) = J(x + \delta x) - J(x) = \int_0^1 \{[x(t) + \delta x(t)]^2 + 4[x(t) + \delta x(t)]\}\,dt - \int_0^1 [x^2(t) + 4x(t)]\,dt \tag{11.20a}$$

Combining these integrals and after simplification, we can write

$$\Delta J(x, \delta x) = \int_0^1 \{[2x(t) + 4]\,\delta x(t) + [\delta x(t)]^2\}\,dt \tag{11.20b}$$

After rearrangement the above equation becomes

$$\Delta J(x, \delta x) = \int_0^1 \{[2x(t) + 4]\, \delta x(t)\}\, dt + \int_0^1 [\delta x(t)]^2\, dt \tag{11.20c}$$

Now let us verify that the second integral can be written in the form:

$$\int_0^1 [\delta x(t)]^2\, dt = g(x, \delta x)\, \| \delta x \| \tag{11.21a}$$

and that
$$\lim_{\| \delta x \| \to 0} \{g(x, \delta x)\} = 0 \tag{11.21b}$$

Since x is a continuous function, let

$$\| \delta x \| \equiv \max_{0 \le t \le 1} \{ | \delta x(t) | \} \tag{11.22}$$

Multiplying the left side of (11.21a) by $\| \delta x \| / \| \delta x \|$ we get

$$\frac{\| \delta x \|}{\| \delta x \|} \cdot \int_0^1 [\delta x(t)]^2\, dt = \| \delta x \| \cdot \int_0^1 \frac{[\delta x(t)]^2}{\| \delta x \|}\, dt\ ; \tag{11.23}$$

the right side of Equation (11.23) follows because $\| \delta x \|$ does not depend on t. Comparing (11.16) with (11.23), we observe that

$$g(x, \delta x) = \int_0^1 \frac{[\delta x(t)]^2}{\| \delta x \|}\, dt. \tag{11.24a}$$

Now writing $[\delta x(t)]^2$ as $| \delta x(t) | \,.\, | \delta x(t) |$ gives

$$\int_0^1 \frac{| \delta x(t) | . | \delta x(t) |}{\| \delta x \|}\, dt \le \int_0^1 | \delta x(t) |\, dt, \tag{11.24b}$$

because of the definition of the norm of δx, which implies that $\| \delta x \| \le | \delta x(t) |$ for all $t \in [0, 1]$. Clearly, if $\| \delta x \| \to 0$ as $\| \delta x(t) \| \to 0$; for all $t \in [0, 1]$, and hence

$$\lim_{\| \delta x \| \to 0} \left\{ \int_0^1 | \delta x(t) |\, dt \right\} = 0$$

We have succeeded in verifying that the increment can be written in the form of Equation (11.16) and that $g(x, \delta x) \to 0$ as $\| \delta x \| \to 0$; therefore, the variation of J is

$$\delta J(x, \delta x) = \int_0^1 \{[2x(t) + 4]\, \delta x(t)\}\, dt. \tag{11.25}$$

This expression can also be obtained by formally expanding the integrand of ΔJ in a Taylor series about $x(t)$ and retaining only the terms of first order in $\delta x(t)$.

It is very important to keep in mind that δJ is the linear approximation to the difference in the functional J caused by two comparison curves. If the comparison curves are close ($\| \delta x \|$ small), then the variation should be a good approximation to the increment; however, δJ may be a poor approximation to ΔJ if the comparison curves are far apart. The analogy in calculus is illustrated in Fig. 11.3 where it is seen that df is a good approximation to Δf for small Δt.

In Section 11.4.3, we shall develop a formal procedure for finding variations of functionals.

We now define the extreme value of a function.

Definition 11.9 A function with domain D has a *relative extremum* at the point z^0, if there is an $\varepsilon > 0$ such that for all points z in D that satisfy $\| z - z^0 \| < \varepsilon$, the *increment* of f has the *same sign*.

Further, if $$\Delta f = f(z) - f(z^o) \geq 0, \tag{11.26}$$

then $f(z^0)$ is a relative minimum

and if $$\Delta f = f(z) - f(z^o) \leq 0, \tag{11.27}$$

then $f(z^0)$ is a relative maximum.

If the relations (11.26) is satisfied for arbitrarily large ε then the $f(z^o)$ is a *global or absolute minimum*. Similarly, $f(z^0)$ will be *global or absolute maximum*, if the relation (11.27) is satisfied for large ε.

Next we define the extremum of a functional L which is defined for all functions x in a class Ω.

Definition 11.10 A functional with domain Ω has a *relative extremum* at the point x^o, if there is an $\varepsilon > 0$ such that for all points x in D that satisfy $\| x - x^o \| < \varepsilon$, the *increment* of L has the *same sign*.

Further, if $$\Delta L = f(x) - f(x^o) \geq 0, \tag{11.28}$$

then $f(x^o)$ is a relative minimum

and if $$\Delta L = f(x) - f(x^o) \leq 0 \tag{11.29}$$

then $f(x^o)$ is a relative maximum.

If the relations (11.28) is satisfied for arbitrarily large ε then the $f(x^o)$ is a *global or absolute minimum*. Similarly, $f(x^o)$ will be *global or absolute maximum*, if the relation (11.29) is satisfied for large ε. The value of $L(x^o)$ is known as *extremum* and x^o is called *extremal*.

11.4.2 The Fundamental Theorem of the Calculus of Variations

The fundamental theorem concerning the computation of extreme values of functions is the necessary condition that the differential vanish at the extreme points (excluding the extrema at the boundary points of a closed regions) [35-36]. In variational problems, the analogous theorem is that the variation must be zero on an extremal curve, provided that there are no bounds imposed on the curves. We first state the theorem, then present its proof [37-39].

Let x be a vector function of t in the class Ω and $L(x)$ be a differentiable function of x. Also we assume that the functions in Ω are not constrained by any boundaries.

Theorem 11.1 The fundamental theorem of the calculus of variation is stated below:

If x^o is an extremal, the variation of the functional L must vanish on x^o.

In other words, $\delta L(x^0, \delta x) = 0$ for all admissible δx.

Proof by contradiction : Assume that x^o is an extremal and that $\delta L(x^o, \delta x) \neq 0$. We will show that these assumptions imply that the increment ΔL can be made to change sign in an arbitrarily small neighborhood of x^o.

The increment of the functional is given by :

$$\Delta L\,(x^0, \delta x) = f(x^o + \delta x) - f(x^o) = \delta L(x^o, \delta x) + g(x^o, \delta x).\, \| \delta x \| \tag{11.30}$$

where $g(x^o, \delta x) \to 0$ as $\| \delta x \| \to 0$

Thus, there is a neighborhood $\| \delta x \| < \varepsilon$, where $g(x^0, \delta x)\, .\, \| \delta x \|$ is small enough so that δL dominates the expression for ΔL. Now let us select the variation $\delta x = k\delta x_1$ as shown in Fig. 11.4 for a scalar function, where $k > 0$ and $\| k\, .\, \delta x_1 \| < \varepsilon$.

Suppose that $\delta L(x^0, k \, . \, \delta x_1) < 0$ (11.31)

Since δL is a linear functional of δx, the principle of homogeneity yields

$$\delta L(x^0, k \, . \, \delta x_1) = k \, . \, \delta L \, (x^0, \delta x_1) < 0 \quad (11.32)$$

Next we consider the variation $\delta x = - k\delta x_1$ as shown in Fig. 11.4. Clearly $k. \, \delta x_1 < \varepsilon$ implies that $\| - k \, . \, \delta x_1 \| < \varepsilon$.

Therefore, $\Delta L(x^0, - k \, . \, \delta x_1)$ is of the same sign as the sign of $\delta L(x^0, - k \, . \, \delta x_1)$. Again, using the principle of homogeneity, we obtain

$$\delta L(x^0, - k \, . \, \delta x_1) = - k \, . \, \delta L(x^0, \delta x_1) \quad (11.33)$$

Therefore, since $\delta L(x^0, k \, . \, \delta x_1) < 0$, $\delta L(x^0, - k \, . \, \delta x_1) > 0$, which implies

$$\Delta L(x^0, - \; k \, . \, \delta x_1) > 0 \quad (11.34)$$

So, we have shown that if $\delta L(x^0, \delta x) \neq 0$, in an arbitrarily small neighborhood of x^o,

$$\delta L(x^0, k \, . \, \delta x_1) < 0 \quad (11.35)$$

and $\Delta L(x^0, - k \, . \, \delta x_1) > 0$ (11.36)

which violates the assumption that x^0 is an extremal (vide Definition 11.10).

Therefore, if x^0 is an extremal, it is necessary that

$$\delta L(x^0, \delta x) = 0 \text{ for arbitrary } \delta x. \quad (11.37)$$

The assumption that the functions in Ω are not bounded guarantees that $k \, . \, \delta x_1$ and $- \, k.\delta x_1$ are both admissible variations.

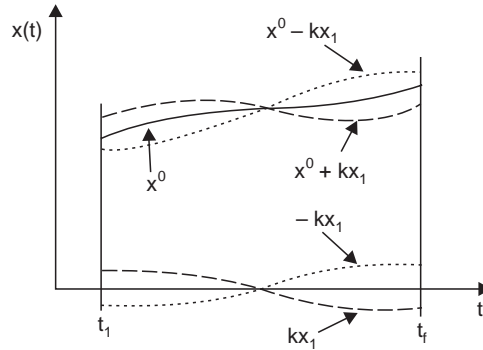



Fig. 11.4 Variation of a function $x(t)$

11.4.3 Extrema of Functionals of a Single Function

In this Section, we shall use the fundamental theorem to determine extrema of functionals, which depends on a single function. Some combinations of end points are shown in Fig. 11.5 with fixed initial time and fixed initial point. There may be other combinations of end conditions encountered in practical optimization problems.

11.4.3.1 Variational Problems and the Euler Equation

Case 1: The Fixed End-point Problem

Let x be a scalar function in the class of functions with continuous first derivatives. It is desired to find the function x^0 for which the functional

$$V(x) = \int_{t_i}^{t_f} L[x(t), \dot{x}(t), t] \, dt = \int_a^b L[x(t), \dot{x}(t), t] dt \quad (11.38)$$

has a relative extremum.

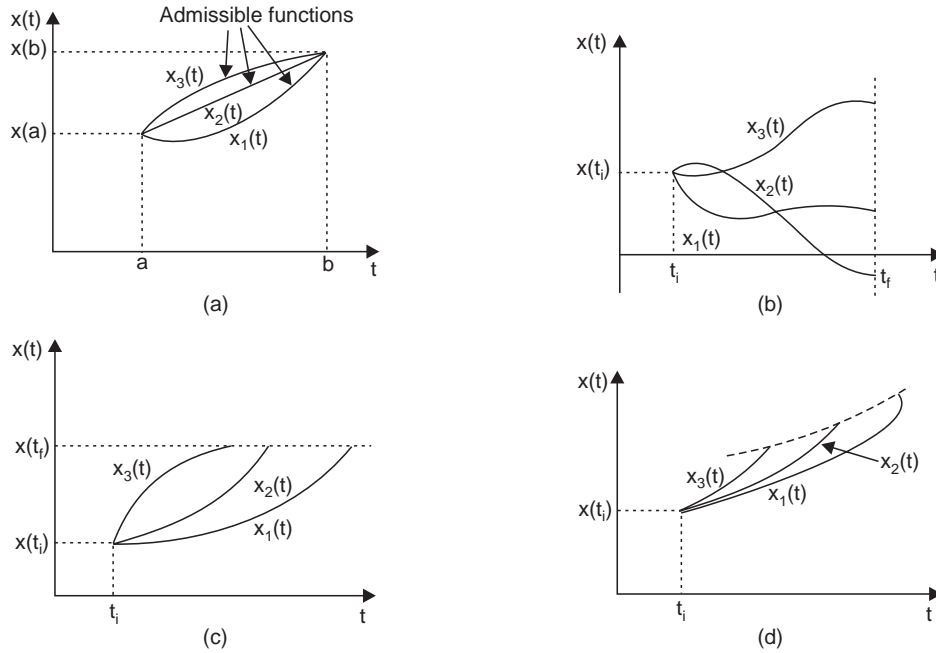



Fig. 11.5 Admissible functions $x(t)$ with fixed initial time t_i and initial point x_i with different combinations of final time and final point; (*a*) both the final time and final point are fixed (*b*) fixed final time and variable final point (*c*) fixed final point and variable final time (*d*) both the final time and final point are variable.

We note that V(x) is a functional of the function x whereas the function $L(x(t), \dot{x}(t), t)$ assigns a real number to the point $(x(t), \dot{x}(t), t)$. The initial time t_i and final time t_f are fixed at **a** and **b** respectively, and the end points of the curve (vide Fig. 11.5a) are also fixed satisfying

$$x(a) = k_1 \text{ and } x(b) = k_2 \text{, where } k_1 \text{ and } k_2 \text{ are constants} \tag{11.39}$$

The functional L is also assumed to possess continuous first and second partial derivatives with respect to all its arguments.

As a first step, we compute the variation of the functional V in order to find the fundamental necessary conditions of Theorem 11.1. Assuming that a extremum of V occurs for the function x, we apply an arbitrary variation δx to the function x to get a new function x_1 given by

$$x_1 = x + \delta x$$

For satisfying the boundary conditions (11.39), we must have

$$\delta x(a) = \delta x(b) = 0 \tag{11.40}$$

Then the corresponding total variation in the functional V is, therefore, given by

$$\Delta V\,(x, \delta x) = \int_a^b L(x + \delta x, \dot{x} + \delta \dot{x}, t)\; dt - \int_a^b L(x, \dot{x}, t)\, dt \tag{11.41}$$

$$= \int_a^b L(x + \delta x, \dot{x} + \delta \dot{x}, t)\; - L(x, \dot{x}, t)\, dt$$

The argument of ΔV shows its dependence on x and δx but not on $\dot{x}$ and $\delta \dot{x}$. This is because of the fact that x and $\dot{x}$, δx and $\delta \dot{x}$ are not independent, they are related as shown below:

$$\dot{x}(t) = \frac{d}{dt}[x(t)], \qquad \delta\dot{x}(t) \; \frac{d}{dt}[\delta x(t)]$$

So, ΔV will be expressed entirely in terms of x and δx.

Expanding the integrand with the help of Taylor's theorem about x and $\dot{x}$, the total variation ΔV becomes

$$\Delta V = \int_a^b \left[\frac{\partial L(x, \dot{x}, t)}{\partial x} \delta x + \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} \delta \dot{x} + \frac{1}{2!} \frac{\partial^2 L(x, \dot{x}, t)}{\partial x^2} \delta x^2 \right.$$

$$\left. + \frac{\partial^2 L(x, \dot{x}, t)}{\partial x \partial \dot{x}} \delta x \, \delta \dot{x} + \frac{1}{2!} \frac{\partial^2 L(x, \dot{x}, t)}{\partial \dot{x}^2} \delta \dot{x}^2 + \ldots \right] dt \tag{11.42}$$

where the L(x, $\dot{x}$, t) terms were cancelled.

The variation of V is found by retaining only the first-order terms in ΔV,

$$\delta V = \int_a^b \left[\frac{\partial L(x, \dot{x}, t)}{\partial x} \delta x + \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} \delta \dot{x} \right] dt \tag{11.43}$$

Now, integrating the second term by parts, one obtains

$$\delta V = \int_a^b \left[\frac{\partial L(x, \dot{x}, t)}{\partial x} - \frac{d}{dt} \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} \right] \delta x \, dt + \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} \delta x \Bigg|_a^b$$

Using the boundary conditions, $\delta x(a) = \delta x(b) = 0$, δV becomes

$$\delta V = \int_a^b \left[\frac{\partial L(x, \dot{x}, t)}{\partial x} - \frac{d}{dt} \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} \right] \delta x \, dt$$

From Theorem 11.1, the necessary condition for x to minimize V yields

$$\delta V = \int_a^b \left[\frac{\partial L(x, \dot{x}, t)}{\partial x} - \frac{d}{dt} \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} \right] \delta x \, dt = 0 \tag{11.44}$$

It can be shown [35, 36] that if a function y is continuous and

$$\int_a^b y(t) \, \delta x(t) \, dt = 0$$

for every function δx that is continuous in the interval [a, b], then y must be zero everywhere in the interval [a, b].

This result is known as the fundamental lemma of the calculus of variation.

Therefore, we can write from Equation (11.44) that for arbitrary variations in x, the integrand must be zero, *i.e.*,

$$\frac{\partial L(x, \dot{x}, t)}{\partial x} - \frac{d}{dt} \frac{\partial L(x, \dot{x}, t)}{\partial \dot{x}} = 0 \tag{11.45}$$

Now, if Equation, (11.45) is not satisfied at some point t_i, then because of the continuity properties of L and x, it must be nonzero and definite with respect to sign in some neighbourhood of t_i. Now choose δx so that it has a constant sign in the neighbourhood and vanishes elsewhere. Then the integral in Equation (11.44) will become nonzero, which is a contradiction of the

fundamental necessary condition. Equation (11.45) is commonly referred to as the *Euler equation*, after Euler, who first obtained it in 1744. The integral curves of the Euler equation are referred to *extremals.*

We summarize the results developed above in the following theorem.

Theorem 11.2 *If a continuously differentiable function x minimizes a functional V of the form*

$$V(x) = \int_a^b L(x,\ \dot{x},\ t)\, dt \tag{11.46}$$

where L has continuous first and second partial derivatives with respect to all its arguments, and if x satisfies the boundary conditions of (11.39), then x satisfies the Euler conditions given by Equation (11.45).

The solution of Euler equation involves the evaluation of two arbitrary constants since the equation is a second order one. These constants are found from the boundary conditions $x(a) = k_1$ and $x(b) = k_2$. It is to be noted that the Euler equation is the necessary condition for extremum of the functional but not sufficient. However, in many cases the physical nature of problem is such that the solution of the Euler equation gives the extremum of the functional.

11.4.3.2 Extrema of Functionals of n Functions

In Section 11.4.3.1 we considered functionals that contained only a single function and its first derivative. In the context of designing optimal controllers for a nth order system, the state variable $x(t)$ is an n-dimensional function. So we are interested to extend the scope of the Theorem 11.2 to the case where the argument of the functional is n independent functions of the form $x(t)' = [x_1(t), x_2(t), x_3(t) .. x_n(t)]$.

For this case the *Theorem 11.2* becomes:

Theorem 11.3 *If a set of continuously differentiable function* **x** *minimizes a functional V of the form*

$$\mathrm{V}(\mathbf{x}) = \int_a^b \mathrm{L}(\mathbf{x},\ \dot{\mathbf{x}},\ t)\, dt \tag{11.47}$$

where L has continuous first and second partial derivatives with respect to all its arguments, and if x satisfies the boundary conditions

$$\mathbf{x}(a) = \mathbf{k_1} \textit{ and } \mathbf{x}(b) = \mathbf{k_2} \tag{11.48}$$

then x satisfies the Euler equations

$$\frac{\partial \mathrm{L}(x, \dot{x}, t)}{\partial x_i} - \frac{d}{dt}\frac{\partial \mathrm{L}(x, \dot{x}, t)}{\partial \dot{x}_i} = 0,\ i = 1, 2, ... n \tag{11.49}$$

The Theorem 11.3 can be proved [36] in the same way as the Theorem 11.2 by varying one of the functions while keeping the others constant.

The Euler equation (11.49) may also be conveniently written in vector notation as shown below, where **x** is an n dimensional vector.

$$\frac{\partial \mathrm{L}(\mathbf{x}, \dot{\mathbf{x}}, t)}{\partial \mathbf{x}} - \frac{d}{dt}\frac{\partial \mathrm{L}(\mathbf{x}, \dot{\mathbf{x}}, t)}{\partial \dot{\mathbf{x}}} = 0 \tag{11.50}$$

It should be noted that the *Theorem 11.2* is proved only for functions possessing continuous first and second derivatives. However, it is possible to show that every solution of the problem possessing only a continuous first derivative, must also satisfy the Euler equation.

11.4.3.3 Variable End Point Problems

We have found the necessary condition to be satisfied by the extremal of a functional with fixed initial time and final time together with fixed initial and final points. But, there are many practical situations where the initial time t_i and initial point x_i are fixed with all possible combinations of unspecified final time t_f and final point $x(t_f)$. We consider below three cases of variational problem for the functional of a single function initially, which is extended subsequently to n functions.

A. Functional with Single Function as Argument

We are interested to find the necessary conditions that must be satisfied by an extremal for a functional of the form

$$V(x) = \int_{t_i}^{t_f} L(x(t),\ \dot{x}(t), t)\, dt\ ; \tag{11.51}$$

where the initial time t_i and initial point $x(t_i) = x_i$ are specified and

Case 2: The final time t_f is specified and final point $x(t_f)$ is free, [Fig. 11.5 (*b*)]

Case 3: The final time t_f is free and final point $x(t_f)$ is specified, [Fig. 11.5 (*c*)]

Case 4: Both final time t_f and final point $x(t_f)$ are free, [Fig. 11.5 (*d*)]

The problems posed in cases 2 and 3, as well as, case 1 are special cases of the problem in case 4. We considered case 1, the simplest of the four cases, in Section 11.4.3.1 and it was found that the extremal function must satisfy the Euler condition for the maximization or minimization of the functional (11.51). The variation of the functional in Equation (11.51) can be found for all the remaining three cases by following a procedure similar to that of Section 11.4.3.1. The necessary conditions can be found by setting this variation of the functional to zero while satisfying the boundary conditions posed under the different cases. We shall skip the detailed derivations in this introductory text and present the final result for the most general case (case 4). The results for case 2 and case 3 can be found as special case of the general case.

Case 4: In Fig. 11.6 we depict an extremal function x^0 and an admissible curve x for specified initial time t_i and initial point x_i. The final time t_f and final point $x(t_f)$ are free. It is to be noted that $\delta x(t_f)$ is the difference in ordinates at $t = t_f$ and δx_f is the difference in ordinates of the end points of the two curves. In general, $\delta x(t_f) \neq \delta x_f$ as is evident in Fig. 11.6.

It can be shown [37] that the variation of the functional, which must vanish on the extremal, is given by

$$\delta V(x^o, \delta x) = 0 = \left[\frac{\partial L}{\partial \dot{x}}(x^o(t_f), \dot{x}^o(t_f)\right] \delta x_f$$

$$+ \left\{L(x^o(t_f), \dot{x}^o(t_f), t_f) - \left[\frac{\partial L}{\partial \dot{x}}(x^o(t_f), \dot{x}^o(t_f), t_f)\right]\dot{x}^o(t_f)\right\} dt_f$$

$$+ \int_{t_i}^{t_f} \left\{\frac{\partial L}{\partial x}(x^o(t_f), \dot{x}^o(t), t) \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{x}}(x^o(t), \dot{x}^o(t), t)\right]\right\} \delta x(t)\, dt \tag{11.52}$$

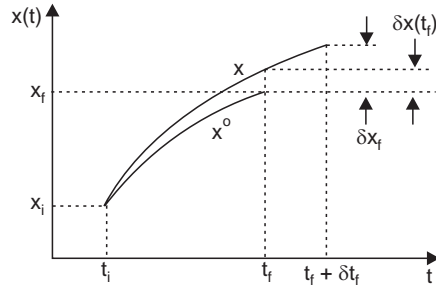



Fig. 11.6 Extremal curve and associated quantities for variable end problem

As a necessary condition, the Euler equation must be satisfied, so the integral is zero as in case 1. The necessary condition on the free terminal points depends on the nature of the end point constrains. Out of numerous end point conditions encountered in practice, only two are illustrated below.

(*a*) The final time t_f and final point $x(t_f)$ are unrelated. In this case, δx_f and δt_f are independent of each other and are arbitrary, so their coefficients must each be zero. Then from Equation (11.52) we get,

$$\frac{\partial L}{\partial \dot{x}}(x^o(t_f), \dot{x}^o(t_f), t_f) = 0, \tag{11.53}$$

and

$$L(x^o(t_f), \dot{x}^o(t_f), t_f) - \left[\frac{\partial L}{\partial \dot{x}}(x^o(t_f), \dot{x}^o(t_f), t_f)\right] . \dot{x}^o(t_f) = 0 \tag{11.54}$$

So, from Equations (11.53) and (11.54) we get the necessary condition as

$$L(x^o(t_f), \dot{x}^o(t_f), t_f) = 0 \tag{11.55}$$

(*b*) The final time t_f and final point $x(t_f)$ are related. The final value of x, for instance, may be constrained to lie on a specified moving point, $\theta(t)$ as given below:

$$x(t_f) = \theta(t_f) \tag{11.56}$$

In such situation the difference in end points δx_f is related to δt_f by the equation

$$\delta x_f \equiv \frac{d\theta}{dt}(t_f)\,\delta t_f \tag{11.57}$$

Using this value of δx_f into Equation (11.52) and collecting terms and in view of arbitrariness of δt_f we get the necessary condition:

$$\left[\frac{\partial L}{\partial \dot{x}}(x^o(t_f), \dot{x}^o(t_f), t_f)\right]\left[\frac{d\theta}{dt}(t_f) - \dot{x}^o(t_f)\right] + L(x^o(t_f), \dot{x}^o(t_f), t_f) = 0 \tag{11.58}$$

The Equation (11.58) is called the *transversality condition.*

Integrating the Euler equation in either of the two cases gives a solution $x^o(k_1, k_2, t)$ where k_1 and k_2 are constants of integration. The constants k_1, k_2 , and the unknown final time t_f can then be determined from $x^o(k_1, k_2, t_i) = x_i$ and Equations (11.53) and (11.55) if $x(t_f)$ and t_f are unrelated, or Equations (11.56) and (11.58) if $x(t_f)$ and t_f are related. We illustrate the application of the above equations with the following example.

Example 11.5 It is required to find an extremal curve for the functional

$$\mathrm{V}(x) \int_{t_i}^{t_f} [1 + \dot{x}^2(t)]^{1/2}\, dt \;; \tag{11.59}$$

with given boundary conditions $t_i = 0$, $x(0) = 0$ whereas the final time t_f and final point $x(t_f)$ are free, but $x(t_f)$ is required to lie on the line

$$\theta(t) = -\,2t + 10 \tag{11.60}$$

The functional V(x) is the length of the curve x ; thus, the function that minimizes V is the shortest curve from the origin to the specified line. The Euler equation is

$$\frac{d}{dt}\left[\frac{\dot{x}^o\,(t)}{[1+\dot{x}^2\,(t)]^{1/2}}\right] = 0. \tag{11.61}$$

Performing the differentiation with respect to time and simplifying, we get

$$\ddot{x}^o\,(t) = 0 \tag{11.62}$$

whose solution is of the form : $x^o\,(t) = k_1 t + k_2$ (11.63)

Here, $k_2 = 0$, since $x^o\,(0) = 0$. We use the transversality condition in order to the find value of the constant of integration k_1. Since $x(t_f)$ and t_f are related, we use relation (11.57), to get

$$\frac{\dot{x}^o\,(t_f)}{[1+\dot{x}^{o2}\,(t_f)]^{1/2}}\,[-\,2 - \dot{x}^o\,(t_f)] + [1+\dot{x}^{o2}\,(t_f)]^{1/2} = 0 \tag{11.64}$$

After simplification, the above equation becomes, $-\,2\dot{x}^o\,(t_f) + 1 = 0$ (11.65)

From Equations (11.63) and (11.65) we obtain $k_1 = 1/2$. The value of t_f, found from

$$x^o\,(t_f) = \theta(t_f)$$

$$\frac{1}{2}t_f = -\,2t_f + 10 \quad \Rightarrow \quad t_f = 4 \tag{11.66}$$

Thus, the function that minimizes the functional in (11.58) is given by : $x^0(t) = \frac{1}{2}t$ (11.67)

Since, the product of the slopes of the optimal trajectory in (11.67) and the trajectory $\theta(t)$ in (11.60) equals $-$ 1, the trajectory $x^0(t)$ is perpendicular to the line $\theta(t)$, which is expected as the shortest path from the origin [$x(t) = 0$, $t = 0$] to a straight line is the normal to the line.

We have already pointed out that the Euler condition must be satisfied regardless of the boundary conditions, thus the integral is zero in Equation (11.52). In addition to the Euler condition, the other necessary conditions to be satisfied in the special cases may be found from the results of general case as follows:

Case 1: The final time t_f and final point $x(t_f)$ are specified along with t_i and $x(t_i)$.

Since $x(t_f)$ is fixed, $dt_f = 0$ and $dx_f = dx(t_f) = 0$. Putting these values in Equation (11.52) one gets the Euler condition in Equation (11.45) already derived.

Case 2: The final time t_f is specified and final point $x(t_f)$ is free, together with specified value of t_i and $x(t_i)$.

Since $x(t_f)$ is free and t_f is specified, set $dt_f = 0$, and $dx_f = dx(t_f)$ in Equation (11.52).

Case 3: The final time t_f is free and final point $x(t_f)$ is specified along with specified value of t_i and $x(t_i)$.

Since $x(t_f)$ is specified, set $dx(t_f) = 0$ in Equation (11.52) to get the necessary conditions for extremal functions.

B. Functional with *n* Functions as Argument

The Equation (11.52) can also be extended to the case of functionals with *n function* as argument by treating $\mathbf{x}$, $\dot{\mathbf{x}}$, $\delta\mathbf{x}$, $\delta\dot{\mathbf{x}}$, $\delta\mathbf{x_f}$ and $\frac{\partial L}{\partial \mathbf{x}}$ as *n* dimensional vectors.

By analogy to Equation (11.52), the necessary condition for extremal of functional with *n* functions are :

1. The Euler equation

$$\frac{\partial L}{\partial \mathbf{x}}(\mathbf{x}^o(t), \dot{\mathbf{x}}^o(t), t) - \frac{d}{dt}\left[\frac{\partial L}{\partial \dot{\mathbf{x}}}(\mathbf{x}^o(t), \dot{\mathbf{x}}^o(t), t)\right] = 0 \quad (11.68)$$

and

2. The boundary conditions at the final time

$$\delta V(\mathbf{x}^o, \delta\mathbf{x}) = 0 = \left[\frac{\partial L}{\partial \dot{\mathbf{x}}}(\mathbf{x}^o(t_f), \dot{\mathbf{x}}^o(t_f))\right]' \delta\mathbf{x}_f$$
$$+ \left\{L(\mathbf{x}^o(t_f). \dot{\mathbf{x}}^o(t_f), t_f) - \left[\frac{\partial L}{\partial \dot{\mathbf{x}}^o}(\mathbf{x}^o(t_f), \dot{\mathbf{x}}^o(t_f), t_f)\right]' \dot{\mathbf{x}}^o(t_f)\right\} dt_f \quad (11.69)$$

Note that $\left[\frac{\partial L}{\partial \dot{x}}\right]'$ is the transpose of the vector $\left[\frac{\partial L}{\partial \dot{x}}\right]$ and is equal to the row vector $\left[\frac{\partial L}{\partial \dot{x}_1} \frac{\partial L}{\partial \dot{x}_2} \cdots \frac{\partial L}{\partial \dot{x}_n}\right]$.

Example 11.6 Find an extremal for the functional

$$J(x) = \int_0^{\pi/4} [x_1^2(t) + \dot{x}_1(t)\dot{x}_2(t) + \dot{x}_2^2(t)]\, dt. \quad (11.70)$$

The functions x_1 and x_2 are unrelated and the boundary conditions are specified as:

$$x_1(0) = 2 \;;\; x_1\left(\frac{\pi}{4}\right) = 4;\; x_2(0) = 3 \text{ and } x_2\left(\frac{\pi}{4}\right) \text{ is free.}$$

From relation (11.68), we write the Euler equation as :

$$2x_1^o(t) - \ddot{x}_2^o(t) = 0 \quad (11.71)$$

$$-\ddot{x}_1^o(t) - 2\ddot{x}_2^o(t) = 0 \quad (11.72)$$

Eliminating $\ddot{x}_2^o(t)$ from (11.71) and (11.72) we get

$$\ddot{x}_1^o(t) + 4x_1^o(t) = 0 \quad (11.73)$$

The Equation (11.73) has solution of the form:

$$x_1^o(t) = k_1 \cos 2t + k_2 \sin 2t, \quad \text{where } k_1, k_2 \text{ are constants} \tag{11.74}$$

Therefore, from (11.72) we can write

$$\ddot{x}_2^o(t) = 2k_1 \cos 2t + 2k_2 \sin 2t \tag{11.75}$$

Integrating both sides of Equation (11.75) we get the value of x_2^o

$$x_2^o(t) = -\frac{k_1}{2} \cos 2t - \frac{k_2}{2} \sin 2t + k_3 t + k_4. \tag{11.76}$$

Since, $x_1(t_f)$ is specified, $\delta x_{1f} = \delta x_1(t_f) = 0$. However, $x_2(t_f)$ is free, so $\delta x_2(t_f)$ is arbitrary. Again, since t_f is specified, we also have $\delta t_f = 0$. Substituting these values in Equation (11.69) yields:

$$\left[\frac{\partial L}{\partial \dot{x}_2}\left\{x^o\left(\frac{\pi}{4}\right), \dot{x}^o\left(\frac{\pi}{4}\right)\right\}\right] \delta x_2 \,(t_f) = 0 \tag{11.77}$$

In view of the fact that $\delta x_2(t_f)$ is arbitrary, the condition in Equation (11.77) implies

$$\frac{\partial L}{\partial \dot{x}_2}\left\{x^o\left(\frac{\pi}{4}\right), \dot{x}^o\left(\frac{\pi}{4}\right)\right\} = 0 \tag{11.78}$$

But from Equation (11.70) we get at $t = t_f = \dfrac{\pi}{4}$

$$\frac{\partial L}{\partial \dot{x}_2}\left\{x^o\left(\frac{\pi}{4}\right), \dot{x}^o\left(\frac{\pi}{4}\right)\right\} = \dot{x}_1^o\left(\frac{\pi}{4}\right) + 2\dot{x}_2^o\left(\frac{\pi}{4}\right) \tag{11.79}$$

The values of $\dot{x}_1^o$, $\dot{x}_2^o$ are computed from Equations (11.74) and (11.76) and after substitution of their values, the right hand side of Equation (11.79) equals k_3. So, k_3 is found to be zero in view of Equation (11.78).

Again, from the specified boundary conditions we have

$$x_1(0) = 2 = k_1 \,.\, 1 + k_2 \,.\, 0 \qquad \Rightarrow \quad k_1 = 2;$$

$$x_2(0) = 3 = -\frac{k_1}{2} \,.\, 1 - \frac{k_2}{2} \,.\, 0 + k_3.0 + k_4 \qquad \Rightarrow \quad k_4 = 3 + \frac{k_1}{2} = 4$$

$$x_1\left(\frac{\pi}{4}\right) = k_1.0 + k_2.1 = 4 \qquad \Rightarrow \quad k_2 = 4$$

Substituting values of the constants, the extremal curve is obtained as:

$$\begin{aligned} x_1^o(t) &= 2 \cos 2t + 4 \sin 2t \\ x_2^o(t) &= -\cos 2t - 2 \sin 2t + 4 \end{aligned} \tag{11.80}$$

We have seen that the extremum can be either minimum or maximum. However, in the discussions that follow we shall consider only the *minimization* of a function x and the results can be readily extended to the problem of *maximization*. We will be concerned only with *local or relative minimum*, since every absolute minimum is also a local minimum. For convenience,

it is assumed that x along with its first and higher derivatives are continuous at any minimum. This assumption eliminates the existence of "extraordinary minimum", occuring at the points of discontinuity of either x or $\dot{x}$. It should be noted, however, that all the results derived below require only the existence of the *first two derivatives* of x.

11.4.4 Optimal Control Problem

We shall now consider the synthesis of an optimal control for a dynamic system that minimizes a performance index. We shall employ the results derived through variation calculus in conjunction with another augmented function known as the *Hamiltonian* to solve the *basic optimal-control problem*. In the basic optimal-control problem, the plant is described by a set of differential equations of the form

$$\dot{\mathbf{x}} = f(\mathbf{x}, \mathbf{u}, t) \tag{11.81}$$

The performance of this system is assessed by an integral performance index of the form

$$J = \int_{t_i}^{t_f} L(\mathbf{x}, \mathbf{u}, t)\, dt \tag{11.82}$$

The limits of the integral are the initial time of action t_i and the final time t_f for which control remains active on the plant. The functional $L(x, u, t)$, is taken to be a positive definite function of x, u, and t. So, the performance index J is a monotonically increasing function of t in the interval $\mathbf{t_i \le t \le t_f}$. The designer is required to find the *optimal control* $u(t)$ or the *optimal-control law* $u = k\,[x(t), t]$ which will transfer the states of the dynamic system (11.81) from a given initial state at $\mathbf{x_i = x(t_i)}$ to a final state $\mathbf{x_f = x(t_f)}$ following a trajectory such that the performance index J in Equation (11.82) will be minimum. The trajectory followed by the dynamic system due to the optimal control or the optimal-control law is called the *optimal trajectory*. In what follows when there is no scope of confusion plain letters will stand for vector quantities for aesthetic reasons.

At this stage, we must point out the difference between the optimal control $u(t)$ and the optimal control law $u = k\,[x(t), t]$, especially from the point of view of their real time realization. In optimal control, the control u is available only as a function of time, and their implementation results in an open-loop system, as shown in Fig. 11.7(a). The implementation of the optimal control law $u = k\,[x(t), t]$, on the other hand, results in a closed loop system as shown in Fig. 11.7(b), since the control law is realized by using the system states measured in real time with the help of suitable transducer.

It is worthwhile to recall that the performance of a control system by classical design approach was assessed with respect to its behavior when subjected to typical inputs like step and sinusoidal signal with all the initial conditions set to zero. The performance of the optimal control system, on the other hand, is assessed by its response to initial conditions rather than its response to an input.

In order to apply the results of the variation calculus presented for fixed end problem in Section 11.4.3.2, we note that we cannot apply the results of *Theorem 11.3* in a straight away fashion to the basic optimal-control problem due to the following differences in the statement of the problems. Even though, both the problems involve the minimization of a functional, and have fixed end points, they differ on the following counts in the problem formulation:

(1) the appearance of the control vector u(t) and (2) the addition of the plant Equation (11.81)

Now the presence of the control vector can be handled by considering the input as part of the state vector and augmenting the n-dimensional state vector to an $(n + r)$-dimensional state vector as

$$x'(t) = [x_1, x_2, ..., x_n, u_1, u_2, ..., u_r] \tag{11.83}$$

This removes one of the obstacles for using the results of Section 11.4.3.2 to the basic optimal-control problem.

However, a completely different problem is posed by the presence of the plant dynamics in basic optimal-control problem.

We know from the solutions of a state equation that if $u(t)$ is specified for the interval $t_i \le t \le t_f$ along with $x(t_i)$, then the value of the state $x(t)$ is uniquely determined for the interval $t_i \le t \le t_f$. On the other hand, if $x(t_i)$ and $\dot{x}(t)$ are known for $t_i \le t \le t_f$, then $u(t)$ is also determined uniquely. In view of this, the plant equations act as constraints on the $x(t)$ and $u(t)$ vectors. Hence the elements $x(t)$ and $u(t)$, cannot be varied independently, as demanded by the proof of *Theorem 11.3*.

One way out to this problem is to express u in terms of x, $\dot{x}$, and t, that is,

$$u(t) = g(x, \dot{x}, t) \tag{11.84}$$

So, $u(t)$ may be eliminated from the expression of the performance index J by using the value from (11.84) that must satisfy the state equation. Therefore, the performance index becomes

$$J = \int_{t_i}^{t_f} L[x, g(x, \dot{x}, t), t]\, dt = \int_{t_i}^{t_f} L'(x, \dot{x}, t), dt \tag{11.85}$$

The basic optimal-control problem then becomes identical to the problem of Section 11.4.3.2, and $\mathbf{x}(t)$ must therefore satisfy the Euler equations (11.49). Once $\mathbf{x}(t)$ is known, $\dot{x}(t)$ may be found and $u(t)$ computed from Equation (11.84).

Fig. 11.7 (a) Open-loop optimal-control problem (b) Closed-loop optimal-control problem

The Lagrange multipliers method can be employed to satisfy the plant-equation constraints, but when it is applied in the variational case the Lagrange constants become functions. This is expected, since in transferring from the ordinary maxima minima of functions to functionals, points have been transformed to functions.

Defining an augmented functional H, known as *Hamiltonian* as given below

$$H(x, u, p, t) = p' f(x, u, t) + L(x, u, t) \tag{11.86}$$

where
$$p'(t) = [p_1(t)\, p_2(t)\, p_3(t)\, ...\, p_n(t)] \tag{11.87}$$

is the set of Lagrange multipliers and $L(x, u, t)$ is the integrand of the performance index of (11.82), we can state the fixed point optimal control problem as:

Theorem 11.4 The function $x(t)$ minimizes the functional of the form

$$V(x) = \int_{t_i}^{t_f} H(x, u, p, t)\, dt \tag{11.88}$$

subject to the constraint
$$\dot{x} = \frac{\partial H(x, u, p, t)}{\partial p} \tag{11.89}$$

if there exists a set of Lagrange multiplier function $p(t)$ given by (11.87) satisfying the equation

$$\dot{p} = -\frac{\partial H(x, u, p, t)}{\partial x} \tag{11.90}$$

and
$$0 = \frac{\partial H(x, u, p, t)}{\partial u} \tag{11.91}$$

It is to be noted that Equation (11.89) is the Equation (11.81) in view of (11.86) and the Equations (11.90) and (11.91) are obtained from the Euler equation where x is given by (11.83).

The optimal control law

$$u^o = u^o\,(x, p, t) \tag{11.92}$$

is obtained as the solution of Equation (11.91), which is then substituted in the Hamiltonian (11.86) to get

$$H^o\,(x, u, p, t) = H(x, u^o\,(x, p, t), p, t] \tag{11.93}$$

Using the above value of H^0, the Equations (11.89) and (11.90) become a set of $2n$ coupled first order differential equations

$$\dot{x} = \frac{\partial H^o(x, u, p, t)}{\partial p} \tag{11.94}$$

$$\dot{p} = -\frac{\partial H^o(x, u, p, t)}{\partial x} \tag{11.95}$$

The optimal trajectory, $x^o(t)$ and $p(t)$ are found by solving (11.94) and (11.95) with $2n$ boundary conditions $x(t_i)$ and $x(t_f)$. The optimal control law is then found by substituting the value of $x^o(t)$ in (11.92).

11.4.5 Pontryagin's Minimum Principle

The Equation (11.91) is obtained by variational method with the assumption that the control u is unbounded. However, in practical situations, the control u is bounded; the output torque of a motor is limited and finite, the thrust of a rocket engine can't exceed an upper limit, the output of a power amplifier is limited to a saturation value and so on. The states are constrained due to safety requirements or structural problems. The speed of an aircraft should be limited to a permissible limit, otherwise the wings may fall off the aircraft body; the current through a heating coil cannot exceed a prescribed limit without damaging the coil. So, the necessary conditions given in (11.91) needs modification when u is bounded. Pontryagin [33] presented a generalized necessary condition than Equation (11.91), which is known as Pontryagin's minimum principle.

Before we formally state the Pontryagin's minimum principle, we present a definition of minimum control u.

The control u^o makes the functional L to attain a relative minimum if

$$L(u) - L(u^0) = \Delta L \geq 0 \tag{11.96}$$

for all admissible controls sufficiently close to u^o. The increment in L can be computed by variational method by taking an incremental change in u from the optimal control as shown below:

$$\Delta L\,(u^o, \delta u) = \delta L\,(u^o, \delta u) + \text{ higher order terms.} \tag{11.97}$$

δL is linear in δu and higher order terms approach zero as $\| \delta u \| \to 0$.

Pontryagin has shown that the necessary condition for u^0 to minimize the functional L is

$$H[x^o, u^o, p^o, t] \le H[x^o, u, p^o, t] \tag{11.98}$$

for all $t \in [t_0, t_f]$ and for *all admissible controls*.

The Equation (11.98), which states that *an optimal control must minimize the Hamiltonian*, is known as Pontryagin's minimum principle.

It is to be noted that, in general, an optimal control must satisfy the Pontryagin's minimum principle but it is not a sufficient condition for optimality; there may be controls that satisfy the minimum principle that are not optimal.

Based on the Pontryagin's minimum principle, we can state optimal control problem as follows:

A control $u^0 \in U$, is required to be found, which causes the system

$$\dot{x} = f(x, u, t) \tag{11.99}$$

to follow an admissible trajectory that minimizes the performance index

$$J(u) = S(x(t_f), t_f) + \int_{t_i}^{t_f} L(x, u, t)\, dt, \tag{11.100}$$

We form the Hamiltonian $H(x, u, p, t) = L(x, u, t) + p'[f(x, u, t)]$ (11.101)

The necessary conditions for u^0 to be an optimal control for $t \in [t_i, t_f]$ are expressed in terms of Hamiltonian:

$$\dot{x}^o = \frac{\partial H^o(x^o, u^o, p^o, t)}{\partial p} \tag{11.102}$$

$$\dot{p}^o = -\frac{\partial H^o(x^o, u^o, p^o, t)}{\partial x} \tag{11.103}$$

$H[x^o, u^o, p^o, t] \le H[x^o, u, p^o, t]$; for all admissible $u(t)$ (11.104)

and
$$\left[\frac{\partial S}{\partial x}(x^o(t_f), t_f) - p^o(t_f)\right]' \delta x_f + \left\{H(x^o(t_f), u^o(t_f), p^o(t_f), t_f) + \frac{\partial S}{\partial t}(x^o(t_f), t_f)\right\} \delta t_f = 0 \tag{11.105}$$

It should be emphasized that

1. $u^o(t)$ is a control that causes $H(x^o(t), u(t), p^o(t), t)$ to assume its *global*, or absolute, minimum.

that is,
$$H^o(x, p, t) = H[x, u^o(x, p, t), p, t] = \min u \in \cup U\, H(x, u, p, t) \tag{11.106}$$

When U is unbounded, the minimum of H can be found by setting the partial derivative of H with respect to u equal to zero. So Equation (11.91) is a special case of Equation (11.106).

2. Equations (11.102) and (11.105) constitute a set of *necessary* conditions for optimality; these conditions are not, in general, sufficient.

In addition, the minimum principle, although derived for controls with values in a closed and bounded region, can also be applied to problems in which the admissible controls are not bounded. This can be done by viewing the unbounded control region as having arbitrarily large bounds, thus ensuring that the optimal control will not be constrained by the boundaries. In this case, for $u^o(t)$ to minimize the Hamiltonian it is necessary (but not sufficient) that

$$\frac{\partial H}{\partial u}(x^o(t), u^o(t), p^o(t), t) = 0 \quad (11.107)$$

If Equation (11.106) is satisfied, and the matrix

$$\frac{\partial^2 H}{\partial u^2}(x^o(t), u^o(t), p^o(t), t)$$

is positive definite, this is sufficient to guaranty that the $u^0(t)$ causes H to be a *local* minimum.

Example 11.7 We consider the following system to illustrate the principle of Pontryagin.

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = 2u$$

with the performance index, $J = \int_0^T |u|\,dt$, where U is the values of u constrained as $|u| < 1$.

The Pontryagin function H for this problem is given by

$$H(x, u, p, t) = |u| + p_1x_2 + 2p_2u$$

Now, we observe that to make H minimum it is necessary that u be given by

$$u = \begin{cases} 1 & \text{if } p_2 \le -1 \\ 0 & \text{if } -1 < p_2 < 1 \\ -1 & \text{if } p_2 \ge 1 \end{cases}$$

Therefore, H is found as

$$H(x, p, t) = \begin{cases} 1 + p_1x_2 + 2p_2 & \text{if } p_2 \le -1 \\ p_1x_2 & \text{if } -1 < p_2 < 1 \\ 1 + p_1x_2 - 2p_2 & \text{if } p_2 \ge 1 \end{cases}$$

This example reveals the important features of Pontryagin approach for the case of bounded control region. It gives the form of the optimal control, even though the exact time response of the optimal control is not known. For this example, the control consists of a sequence of three values – 1, 0, and 1, that is, the system is subjected to maximum available input, either in the negative or positive direction or allowed to coast without any control.

Example 11.8 Consider a first order system

$$\dot{x} = -\sqrt{2}x + \sqrt{2}u$$

for which an optimal controller is to be found by Pontryagin minimum principle that will transfer the initial state $x(0) = 1$ at $t = 0$, to a final state $x(1) = 0$ at $t = 1$, such that the following performance index is minimized:

$$J = \int_0^1 (x^2 + u^2)dt$$

In this example, it is assumed that the control u is unconstrained.

As a first step, we form the Pontryagin function $H(x, u, p, t) = p(-\sqrt{2}x + \sqrt{2}u) + x^2 + u^2$

Now by setting $\frac{\delta H(x, u, p, t)}{\delta u} = \sqrt{2}p + 2u = 0$, we get $u^o = -\frac{p}{\sqrt{2}}$. The optimal value of H is found by substituting the value of u^o in H.

$$H^o(x, p, t) = -\sqrt{2}px - p^2 + x^2 + \frac{p^2}{2} = -\sqrt{2}px - \frac{p^2}{2} + x^2$$

Therefore, from Equations (11.94) and (11.95), we can write

$$\dot{x} = \frac{\delta H^o(x, p, t)}{\delta p} = -\sqrt{2}x - p$$

and
$$\dot{p} = -\frac{\delta H(x, p, t)}{\delta x} = -2x + \sqrt{2}p$$

From the above pair of coupled equations, we can solve for $x(t)$ and $p(t)$. Differentiating both sides of the first equation with respect to time and using the value of $\dot{p}$, we get

$$\ddot{x} = -\sqrt{2}\dot{x} - \dot{p} = 2x + \sqrt{2}p + 2x - \sqrt{2}p = 4x.$$

So the eigenvalues are given by $a_{1,2} = \pm 2$.

Therefore, the solution of $x(t)$ is of the form $x(t) = k_1e^{-2t} + k_2e^{2t}$, where k_1, k_2 are constants.

The values of the constants k_1, k_2 can be found from the boundary conditions on $x(t)$, which are $x(0) = 1$ at $t = 0$, and state $x(1) = 0$ at $t = 1$.

Substituting these boundary values in the solution of $x(t)$, we get

$$k_1 + k_2 = 1 \quad \text{and} \quad k_1e^{-2} + k_2e^2 = 0$$

Solving the above pair of equations we find k_1 and k_2 as

$$k_1 = \frac{1}{1 - e^{-4}} \quad \text{and} \quad k_2 = \frac{1}{1 - e^4}$$

These values of k_1 and k_2 can be substituted to find the optimal trajectory $x^0(t)$ and its derivative can also be computed.

Again, from the state equation we note that $p = -\dot{x} - \sqrt{2}x$

So, putting the values of $x(t)$ and $\dot{x}(t)$ in the above equation we get

$$p = 2k_1e^{-2t} - 2k_2e^{2t} - \sqrt{2}k_1e^{-2t} - \sqrt{2}k_2e^{2t}$$
$$= k_1(2 - \sqrt{2})e^{-2t} - k_2(2 + \sqrt{2})e^{2t}$$

Substituting the value of k_1 and k_2, we get

$$p(t) = \frac{(2 - \sqrt{2})}{1 - e^{-4}}e^{-2t} - \frac{(2 + \sqrt{2})}{1 - e^4}e^{2t}$$

Therefore, the optimal control u^o is finally found as

$$u^o(t) = -\frac{p}{\sqrt{2}}$$

Substituting the value of p we get,

$$u^o(t) = -\left(\frac{\sqrt{2} - 1}{1 - e^{-4}}\right)e^{-2t} + \left(\frac{\sqrt{2} + 1}{1 - e^4}\right)e^{2t}$$

11.5 THE LQ PROBLEM

The Pontryagin approach to the optimal-control problem involves the solution of a two-point boundary-value problem and results in an open loop controller, whereas, the alternative, known

as the Hamilton-Jacobi approach yields a closed-loop solution in the form of an optimal-control law $u^o(x, t)$.

However, the solution of the nonlinear partial differential Hamilton-Jacobi equation is so difficult that it has been accomplished only for a few special cases. Hence one must often return to the Pontryagin approach with its undesirable open-loop answer in order to be able to achieve a solution. This added difficulty of the Hamilton-Jacobi approach is expected since we seek the solution for a whole family of initial conditions by state feedback instead of only one specific set of initial conditions as is done in the case of Pontryagin approach.

There is, however, one class of problems of significant practical importance known as linear quadratic (LQ) problems, for which it is possible to solve the Hamilton-Jacobi equation in a reasonably simple manner.

The problem of interest is the optimal control of the linear system

$$\dot{x}(t) = A(t)x(t) + B(t)u(t) \tag{11.108}$$

$$x(t_i) = x_i$$

with given initial state x_i and time t_i, a control is to be found which minimizes the performance index

$$J = \frac{1}{2}\left\{\int_{t_i}^{t_f} [x'(t)Q(t)x(t) + u'(t)R(t)u(t)]\, dt + x'(t_f)S_f x(t_f)\right\} \tag{11.109}$$

The final time t_f is given, while no constraints are imposed on the final state $x(t_f)$. The matrices $A(t)$, $B(t)$, $Q(t)$ and $R(t)$ in Equations (11.108) and (11.109) are continuously differentiable functions and $Q(t) = Q'(t) \geq 0$, $R(t) = R'(t) > 0$, $\forall\, t\, \varepsilon\, [t_i, t_f]$, $S_f = S_f' \geq 0$ [here $Q'(t)$ stands for the transpose of $Q(t)$].

The solution of the linear control problem is found from two different approaches. First, the Hamilton-Jacobi approach is directly applied to the above problem to develop a matrix form of solution in the time domain known as the *matrix Riccati equation*. Another approach is to employ some concepts from the second method of Lyapunov, to derive the same matrix result.

It is assumed here that all the state variables are available for measurement, which in many practical systems is not justified, and one must use observers for synthesizing the unmeasurable states.

11.5.1 The Hamilton-Jacobi Approach

The Lagrange multiplier-Pontryagin method is concerned with solution of the basic optimal-control problem, in which the system is described by

$$\dot{x} = f(x, u, t) \tag{11.110}$$

with the performance index $J = \frac{1}{2}\int_{t_i}^{t_f} L(x, u, t)\, dt$ (11.111)

The Lagrange multiplier-Pontryagin method requires the solution of a two-point boundary problem, which is not only computationally involved, it yields only open-loop control system. The Hamilton-Jacobi approach is not free from computational difficulties, but fortunately, for a class of problems the Hamilton-Jacobi equation can be solved in a straightforward and, in some cases, even simple manner. Besides, the resulting control law is a function of the state variables, indicating that it yields closed-loop control system.

Since we are interested in a closed-loop solution, we begin the development by assuming that the control u is a function of states and time of the form $u^o(x, t)$. The explicit dependence of u^o on t is because of the fact that the plant or the performance index may be time-varying, and is not dependent on the initial states alone, as in the open-loop solution.

Next, we define a scalar function $V(x, t)$ as the minimum value of the performance index for an initial state x_i at time t_i.

So we write, $$V(x,t) = \frac{1}{2}\int_{t_i}^{t_f} L\,[x(\tau), u^o(x,\tau), \tau]\, d\tau$$

Therefore, we can say that $V(x, t)$ is the value of the performance index when evaluated along an optimal trajectory which begins at $x(t_i)$. The function $V(x, t)$ may be treated as a Lyapunov function, whose total time derivative is given by

$$\dot{V}(x,t) = -\frac{1}{2}\, L\,[x, u^o(x, t), t] \tag{11.112}$$

The negative sign on the right hand side appears because the differentiation is with respect to the lower limit of the integral. Again, applying the chain rule for differentiation yields the following expression for $\dot{V}(x,t)$:

$$\dot{V}(x,t) = \frac{1}{2}\sum_{i=1}^{n} \frac{\partial V(x,t)}{\partial x_i} f_i\,[x, u^o(x,t), t] + \frac{1}{2}\frac{\partial V(x,t)}{\partial t}$$

$$= \frac{1}{2}\nabla V'(x,t) f\,[x, u^o(x,t), t] + \frac{1}{2}\frac{\partial V(x,t)}{\partial t} \tag{11.113}$$

where the optimal control has been substituted for u. Equating these two expressions for $\dot{V}(x, t)$, we get

$$\nabla V'(x,t) f\,[x, u^o(x,t), t] + \frac{\partial V(x,t)}{\partial t} = -L[x, u^o(x, t), t] \tag{11.114}$$

or $$\nabla V'(x,t) f\,[x, u^o(x,t), t] + \frac{\partial V(x,t)}{\partial t} + L[x, u^o(x, t), t] = 0 \tag{11.115}$$

The Equation (11.115) is known as *Hamilton-Jacobi equation* with optimal Hamiltonian function given by

$$H = L[x, u^o(x, t), t] + \nabla V'(x, t)\, f\,[x, u^o(x, t), t] \tag{11.115}$$

We note that $\nabla V'$ is of the same form as λ' in the Equation (11.86) for the Pontryagin function.

If $u^o(x, t)$ were known, this expression (11.115) would represent a partial differential equation for $V(x, t)$. On the other hand, if $V(x, t)$ were known, it would be possible to solve the equation for $u^o(x, t)$. The actual manner in which we shall use Equation (11.115) is a combination of these two possibilities. The control law $u^o(x, t)$ will be expressed as a function of ∇V, x, and t, and this relation is then substituted for u^o in Equation (11.115). This results in a partial differential equation which may be solved for $V(x, t)$. From known $V(x, t)$, $u^o(x, t)$ may be found.

11.5.2 The Matrix Riccati Equation

The previous section developed the Hamilton-Jacobi approach to the basic optimal-control problem, in which the system is described by Equation (11.110) and the performance index by

Equation (11.111). The linear control problem is a subset of the general optimal-control problem in which the system is described by Equation (11.108) and the performance index by Equation (11.109) and u is unconstrained.

11.5.3 Finite Control Horizon

The following result holds for the LQ problem over a finite horizon.

Theorem 11.5 *For the dynamic system(11.108), any initial state* x_i *can be transferred to an arbitrarily chosen final state* $x(t_f)$ *in a finite control interval* $[t_i, t_f]$ *by the control law*

$$u^o(x, t) = - R^{-1}(t)B'(t)S(t)x \qquad (11.116)$$

that minimizes the performance index in Equation (11.109), where the matrix S *is the solution of the differential Riccati equation*

$$\dot{S}(t) = - S(t)A(t) - A'(t)S(t) + S(t)B(t)R^{-1}(t)B'(t)S(t) - Q(t) \qquad (11.117)$$

with boundary condition $S(t_f) = S_f$ (11.118)

Further, the minimal value of the performance index is

$$J^o(x_i, t_i) = \frac{1}{2} x_i' S(t_i) x_i$$

Proof. The first step of the proof is the formation of the Hamiltonian function, which for the above problem becomes

$$H(x, u, \nabla V, t) = \nabla V' [A(t)x + B(t)u] + \frac{1}{2} [x'Q(t)x + u'R(t)u] \qquad (11.119)$$

We note that ∇V plays the same role in the Hamiltonian function, the Lagrange multiplier function $p(t)$ played in the Equation (11.86). [Often the arguments of functions are omitted for brevity].

Upon setting $\partial H/\partial u = 0$, we obtain

$$B'(t)\nabla V + R(t)u = 0$$

Therefore $u^o(x, t)$ is given by

$$u^o = - R^{-1}(t)B'(t)\nabla V(x, t) \qquad (11.120)$$

The matrix $R(t)$ must be positive definite so that R^{-1} always exists. In a physical sense, if R were only semi definite, some elements of the control vector could become infinite without affecting the value of the performance index.

Substituting the optimal value of u^o into the expression (11.119), we get the optimal Hamiltonian function as:

$$H^o(x, \nabla V, t) = \nabla V'A(t)x - \nabla V' B(t)R^{-1}B'(t)\nabla V + \frac{1}{2} x'Q(t)x + \frac{1}{2}\nabla V'B(t)R^{-1}(t)R(t)R^{-1}(t)B'(t)\nabla V$$

$$= \nabla V'A(t)x - \frac{1}{2}\nabla V' B(t)R^{-1}(t)B'(t)\nabla V + \frac{1}{2}x' Q(t)x$$

The Hamilton-Jacobi equation (11.115) then becomes

$$\nabla V'A(t)x - \frac{1}{2}\nabla V'B(t)R^{-1}(t)B'(t)\nabla V + \frac{1}{2}x'Q(t)x + \frac{\partial V}{\partial t} = 0 \qquad (11.121)$$

In order to find the optimal-control law $u^o(x, t)$, the above partial differential equation must be solved. The approach that will be adopted here for getting a solution is to assume that the solution is in time-varying quadratic form and then by showing that such a solution satisfies the Hamilton-Jacobi equation.

We assume that $V(x, t)$ is given by

$$V(x, t) = \frac{1}{2}x'S(t)x = \frac{1}{2}\left[\int_t^T (x'Q(\tau)x + u'R(\tau)u)\right]d\tau \tag{11.122}$$

where $S(t)$ is a positive definite symmetric matrix for all $t < T$ and $S(T) = 0$. This latter condition is due to the boundary condition on $V(x, t)$, since with $x(T)$ free, $V(x,T)$ is independent of x in view of the following fact:

$$V(x, T) = \frac{1}{2}\left[\int_T^T (x'Q(t)x + u'R(t)u)\,dt\right] = 0$$

The requirement that $S(t)$ be positive definite is necessary since the value of the integrand of the performance index is a positive definite function.

From the assumed structure of $V(x, t)$ in relation (11.122), ∇V and $\partial V/\partial t$ are found as:

$$\nabla V = S(t)\,x \tag{11.123}$$

and

$$\frac{\partial V}{\partial t} = \frac{1}{2}x'\dot{S}(t)x \tag{11.124}$$

Substituting these results into the Hamilton-Jacobi equation, we get

$$x'S(t)A(t)x - \frac{1}{2}x'S(t)BR^{-1}B'S(t)x + \frac{1}{2}x'Q(t)x + \frac{1}{2}x'\dot{S}(t)x = 0$$

or

$$x'\left[S(t)A(t) - \frac{1}{2}S(t)BR^{-1}B'S(t) + \frac{1}{2}Q(t) + \frac{1}{2}\dot{S}(t)\right]x = 0 \tag{11.125}$$

This equation can be satisfied for arbitrary x if the quantity inside the brackets is equal to zero. However, this statement needs some amendment. Since the Equation (11.125) is quadratic in nature, it is necessary that only the symmetric part of the bracketed expression be zero. So, for arbitrary x, the symmetric matrix $S(t)$ in (11.125) satisfies the differential equation

$$\dot{S}(t) + S(t)A(t) + A'(t)S(t) - S(t)B(t)R^{-1}(t)B'(t)S(t) + Q(t) = 0 \tag{11.126}$$

subject to the boundary condition $S(T) = 0$.

It is to be noted that the symmetric part of the matrix product

$$S(t)A(t) = \frac{1}{2}[S(t)A(t) + A'(t)S(t)], \text{ in view of } S' = S.$$

This completes the proof. The Equation (11.126) is known as the *matrix Riccati equation.*

Even though general methods are available for obtaining an analytic solution for $S(t)$, they are unwieldy for any systems of order higher than second. On the other hand, it is convenient to obtain a numerical solution for $S(t)$ on a digital computer by integrating the Riccati equation backward in time from the known terminal condition over the time interval of interest.

Once $S(t)$ is known for $0 \le t \le T$, the optimal-control law may be obtained from Equations (11.120) and (11.123) as

$$u^o(x, t) = -R^{-1}B'S(t)x \tag{11.127}$$

This is often written in the form

$$u^o(x, t) = -\,K'(t)x \qquad (11.128)$$

where $$K(t) = S(t)\,BR^{-1} \qquad (11.129)$$

The elements of the matrix K(t) are referred to as the *feedback coefficients*, which is similar in structure to the feedback matrix in pole assignment design. However, the design philosophy is different and if the final time t_f in relation (11.109) is finite, K(t) will be functions of time and linear combination of the state x(t). This result is important since it indicates that in an optimal system all the state variables are to be fed back, in contrast to only the output, as is customary in classical control theory. A graphical representation of this optimal system is presented in Fig. 11.8.

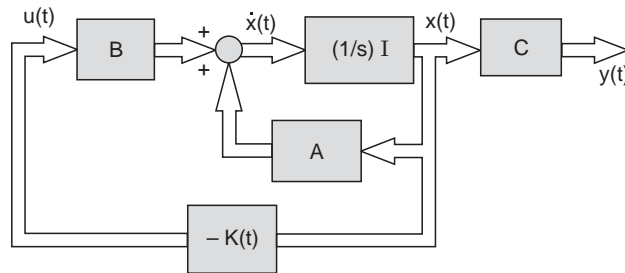



Fig. 11.8 Structure of feedback control

11.5.4 Linear Regulator Design (Infinite-Time Problem)

In a regulator problem, the feedback control system is required to operate around a fixed set point which is constant and the terminal time is infinite. In this case the differential matrix Riccati equation (11.126) becomes an algebraic equation with $\dot{S} = 0$. The solution becomes easier as demonstrated by the following illustrative example.

Example 11.9 Find the optimal-feedback coefficients for the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$

if the performance index is $J = \int_0^\infty (x_1^{\,2} + x_2^{\,2} + 0.2u^2)\,dt$

Here the matrices Q and R are, respectively Q = [1 0;0 1], R = 0.2. We can solve the Equation (11.126) and find the optimal control law from (11.128) and (11.129). It can be found by the following MATLAB script file:

```
> A = [0 1 ; – 2 – 3];
> B = [0 ; 1];
> Q = [1 0 ; 0 1];
 > R = 0.2
> [K,S,E] = lqr (A, B, Q, R);
```

where K is the feedback coefficient and E is the eigenvalue of the closed loop system matrix (A – B*K) and S is the solution of the matrix Riccati equation. By typing >> *help lqr* in the MATLAB command window one can see the format of the command *lqr*.

For this system the feedback coefficients are found as k = [1.0000 1.0000]

11.6 OPTIMAL CONTROLLER FOR DISCRETE SYSTEM

Consider now a discrete system described as

$$x(k+1) = Ax(k) + Bu(k) \tag{11.130}$$

with known initial condition $x(0)$. The design objective is to find a control $u^o(k)$ so that the performance index

$$J_N = \frac{1}{2} x'(N)S_N\, x(N) + \frac{1}{2}\sum_{k}^{N-1} [x'(k)Qx(k) + u'(k)Ru(k)] \tag{11.131}$$

is minimized. We form the Hamiltonian function (here $p(k)$ plays the role of Lagrange multiplier)

$$H(k) = H[x(k), p(k+1), u(k)] = 1/2\, x'(k)Qx(k) + 1/2\, u'(k)Ru\,(k) + p'(k+1)Ax(k) + p'(k+1)Bu(k) \tag{11.132}$$

The necessary conditions for J_N to be an extremum are given by

$$\frac{\partial H^o(k)}{\partial x^o(k)} = p^o(k) = Qx^o(k) + A'p^o(k+1) \tag{11.133}$$

$$\frac{\partial H^o(k)}{\partial p^o(k+1)} = x^o(k+1) = Ax^o(k) + Bu^o(k) \tag{11.134}$$

and
$$\frac{\partial H^o(k)}{\partial u^o(k)} = Ru^o(k) + B'p^o(k+1) = 0 \tag{11.135}$$

In this case, since x^o (N) is not specified, the application of transversality condition yields:

$$\frac{\partial}{\partial x(N)}\left[\frac{1}{2} x'(N)\, S_N\; x(N)\right] = S_N x(N) = P(N) \tag{11.136}$$

The optional control is obtained from Equation (11.135)

$$u^o(k) = -R^{-1}B'p^o(k+1) \tag{11.137}$$

Substituting the value of $u^o\,(k)$ in (11.134) and rearranging (11.133), we get

$$x^o(k+1) = Ax^o(k) - BR^{-1}B'p^o(k+1) \tag{11.138}$$

$$A'p^o(k+1) = p^o(k) - Qx^o(k) \tag{11.139}$$

The coupled state equation (11.138) and (11.139) are difficult to solve. However, it can be shown that the solution is of the form:

$$p(k) = S(k)\, x(k) \tag{11.140}$$

where $x(k)$ is an $n \times n$ matrix with yet unknown properties except that at $k = N$, from Equation (11.136),

$$S(N) = S_N \tag{11.141}$$

Substituting Equation (11.140) into (11.138) and arranging the terms, we get

$$x^o(k+1) = [I + BR^{-1}\, B'S(k+1)]^{-1}\, Ax^o(k) \tag{11.142}$$

where it is assumed that the inverse of $[I + BR^{-1}\, B'\, S(k+1)]$ exists.

It can be shown that if $S(k+1)$ is positive semi definite, the inverse will exist. From (11.140) and (11.139) we get

$$A'S(k+1)x^o\,(k+1) = [S(k) - Q]x^o(k) \tag{11.143}$$

Putting the value of $x(k+1)$ from (11.142) into (11.143) we get

$$A'S(k+1)[I + BR^{-1} B'S(k+1)]^{-1}Ax^o(k) = [S(k) - Q]\,x^o(k) \tag{11.144}$$

So, for any value of $x^o(k)$, the following must hold :

$$A'S(k+1)[I + BR^{-1} B'S(k+1)]^{-1} A = S(k) - Q \tag{11.145}$$

The last equation represents a nonlinear matrix difference equation in S(k), which is of the Riccati type and is generally called the discrete Riccati Equation. The $n \; x \; n$ matrix S(k) is hereafter referred to as Riccati gain.

When S(k) is symmetric, there are $n(n+1)$ unknowns. The optimal control is obtained by substituting Equation (11.140) into Equation (11.137) and using (11.142).

$$u^o(k) = -R^{-1} B'S(k+1)[I + BR^{-1} B'S(k+1)]^{-1} Ax^0(k)$$

Since R and S are symmetric matrices, we can write

$$\begin{aligned} u^o(k) &= -[I + R^{-1} B'S(k+1)B]^{-1} R^{-1} B'S(k+1)Ax^0(k) \\ &= -[R + B'S(k+1)B]^{-1} [B'S(k+1)A]\, x^0(k) \\ &= -[R + B'S(k+1)B]^{-1} B'S(k+1)Ax^0(k) \\ &= -\mathbf{K(k)\, x^0\, (k)} \end{aligned} \tag{11.146}$$

where

$$\mathbf{K(k) = [R + B'S(k+1)\, B]^{-1}\, B'S(k+1)A} \tag{11.147}$$

The value of S can be solved recursively from the following equation derived from (11.145) and the identity in (11.149)

$$\mathbf{S(k) = Q + A'S(k+1)\, [A - B\, K(k)]} \tag{11.148}$$

The Equation (11.148) has been derived from Equation (11.145) by using the identity:

$$S(k+1)\,[I + B\,R^{-1}\,B'S(k+1)]^{-1} = S(k+1) - S(k+1)\,B\,[R + B'\,S(k+1)\,B]^{-1}\,B\,S(k+1) \tag{11.149}$$

Using above identity relation (11.145) can be written as

$$A'\{S(k+1) - S(k+1)\,B\,[R + B'S(k+1)\,B]^{-1}\,B\,S(k+1)\}\,A = S(k) - Q$$

$\Rightarrow$ $A'S(k+1)\,A - A'S(k+1)\,B\,K(k) = S(k) - Q$; using the value of K($k$) from (11.147) and hence the result in Equation (11.148).

In order to prove the identity in (11.149), let

$$M = S(k+1)\,[I + BR^{-1}B'\,S(k+1)]^{-1} \tag{11.150}$$

So, the identity in (11.149) becomes after rearrangement

$$S(k+1)\,B\,[R + B'S(k+1)\,B]^{-1}\,B\,S(k+1) = S(k+1) - M \tag{11.151}$$

$$\Rightarrow \quad S(k+1)\,BR^{-1}[I + R^{-1}B'S(k+1)\,B]^{-1}\,BS(k+1) + M = S(k+1)$$

Now post multiplying both sides of Equation (11.150) by $[I + BR^{-1}B'S(k+1)]B$ we get

$$M\,[I + BR^{-1}\,B'\,S(k+1)]\,B = S(k+1)B$$

$$\Rightarrow \quad M\,B + M\,B\,R^{-1}B'S(k+1)B = S(k+1)B$$

$$\Rightarrow \quad M\,B\,R^{-1}\,[R + B'S(k+1)\,B] = S(k+1)\,B$$

Post multiplying both sides of the above relation by $[R + B'S(k+1)\,B]^{-1}\,B\,S(k+1)$ yields

$MBR^{-1}\,BS(k+1) = S(k+1)\,B\,[R + B'S(k+1)\,B]^{-1}\,B\,S(k+1) = S(k+1) - M$, using (11.151).

$$\Rightarrow \quad M\,B\,R^{-1}\,B\,S(k+1) + M = S(k+1)$$

i.e. if $\quad M\,[I + M\,B\,R^{-1}\,B\,S(k+1)] = S(k+1)$

$\Rightarrow$ $M = S(k+1)\,[I + B\,R^{-1}B'S(k+1)]^{-1}$ which was the assumption.

11.6.1 Linear Digital Regulator Design (Infinite-Time Problem)

For infinite time, N = ∞. The performance index in (11.131) is written as:

$$\mathbf{J = \frac{1}{2} \sum_{k=0}^{\infty} [x'(k) Q\, x(k) + u'(k)\, R\, u(k)]}$$

In this case, the terminal cost is eliminated, since N → ∞, the final state x(N) should approach the equilibrium state, which is the origin *i.e.* x(N) = 0.

For the initial time problem, the system [A,B] must be controllable and the system [A, D] must be observable, where $DD^T = Q$.

Then the solution to the infinite time linear regulator problem can be obtained by setting $k \to \infty$.

as $$N \to \infty,\ \lim_{k \to \infty} S(k) = S \tag{11.152}$$

Replacing S(k + 1) and S(k) by S in (11.145), we have

$$A'S[I + BR^{-1} B'S]^{-1} A = S - Q \tag{11.153}$$

and $$u^0(k) = -[R + B'SB]^{-1} B'SAx^0(k) \tag{11.154}$$

$$= -Kx^0(k) \text{ where } K = [R + B'SB]^{-1} B'SA \tag{11.155}$$

The Riccati gain matrix S is obtained as the solution of (11.153).

Example 11.10 Let us consider the discretized version of the continuous system of Example 9.1 in Chapter 9 with sampling period of 0.1 second described by the state equation

$$x(k + 1) = Ax(k) + Bu(k) \tag{11.156}$$

where $$A = \begin{bmatrix} 1 & 0.0824 \\ 0 & 0.6703 \end{bmatrix},\quad B = \begin{bmatrix} 0.4395 \\ 8.2420 \end{bmatrix} \tag{11.157}$$

Given that $x(0) = [1 \quad 1]'$, let us find the optimal control $u(k)$, k = 0, 1, 2, ... 7, such that the performance index

$$J_8 = \sum_{k=0}^{7} [x_1^2(k) + 0.005\, x_2^2(k) + u^2(k)] \tag{11.158}$$

is minimized.

For this problem we identify that R = 2

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 0.01 \end{bmatrix} \quad \text{and} \quad S_f = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{11.159}$$

The Riccati equation is obtained by substituting these parameters into Equation (11.148) to get

$$S(i) = \begin{bmatrix} 2 & 0 \\ 0 & 0.01 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0.0824 & 0.6703 \end{bmatrix} S(i+1) \left[\begin{bmatrix} 1 & 0.0824 \\ 0 & 0.6703 \end{bmatrix} - \begin{bmatrix} 0.4395 \\ 8.2420 \end{bmatrix} K(i) \right] \tag{11.160}$$

The optimal control is $u^o(i) = -K(i)\, x^o(i)$ (11.161)

where K(i) must satisfy Equation (11.147), which, for the present case becomes:

$$K(i) = \left[2 + [0.4395 \quad 8.2420]\, S(i+1) \begin{bmatrix} 0.4395 \\ 8.2420 \end{bmatrix} \right]^{-1} [0.4395 \quad 8.2420]\, S(i+1) \begin{bmatrix} 1 & 0.0824 \\ 0 & 0.6703 \end{bmatrix} \tag{11.162}$$

Starting with the boundary condition,

$$S(8) = S_f = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{11.163}$$

Chapter 11

Equations (11.160) and (11.162) are solved recursively to yield, **backward in time**, the following (see Script_Example11-10)

$$S(7) = \begin{bmatrix} 2 & 0 \\ 0 & 0.01 \end{bmatrix} \qquad K(7) = [0\ 0]$$

$$S(6) = \begin{bmatrix} 3.7480 & 0.1282 \\ 0.1282 & 0.0228 \end{bmatrix} \qquad K(6) = [0.2867 \quad 0.0417]$$

$$S(5) = \begin{bmatrix} 4.3416 & 0.1939 \\ 0.1939 & 0.0311 \end{bmatrix} \qquad K(5) = [0.5201 \quad 0.0743]$$

$$S(4) = \begin{bmatrix} 4.4082 & 0.2021 \\ 0.2021 & 0.0327 \end{bmatrix} \qquad K(4) = [0.5514 \quad 0.0815]$$

$$S(3) = \begin{bmatrix} 4.4220 & 0.2027 \\ 0.2027 & 0.0328 \end{bmatrix} \qquad K(3) = [0.5513 \quad 0.0822]$$

$$S(2) = \begin{bmatrix} 4.4285 & 0.2031 \\ 0.2031 & 0.0329 \end{bmatrix} \qquad K(2) = [0.5516 \quad 0.0823]$$

$$S(1) = \begin{bmatrix} 4.4304 & 0.2032 \\ 0.2032 & 0.0329 \end{bmatrix} \qquad K(1) = [0.5520 \quad 0.0823]$$

$$S(0) = \begin{bmatrix} 4.4308 & 0.2033 \\ 0.2033 & 0.0329 \end{bmatrix} \qquad K(0) = [0.5521 \quad 0.0823]$$

The optimal control and the optimal trajectories are computed by substituting the feedback gains into Equations (11.161) and (11.156). The results are tabulated below:

i	$u^0(i)$	$x_1^0(i)$	$x_2^0(i)$
0	– 0.6344	1.0000	1.0000
1	– 0.0685	0.8036	– 4.5582
2	0.0783	0.3978	– 3.6199
3	0.0726	0.1339	– 1.7811
4	0.0384	0.0190	– 0.5951
5	0.0140	– 0.0131	– 0.0823
6	0.0027	– 0.0138	0.0598
7	0.0004	– 0.0077	0.0626
8	0.0000	– 0.0027	0.0386

For large values of N we can show that the Riccati gain approaches the steady-state solution of

$$S = \begin{bmatrix} 4.4308 & 0.2033 \\ 0.2033 & 0.0329 \end{bmatrix} \tag{11.164}$$

and the constant optimal control is

$$K = [0.5521\ 0.0823]$$

The values of S and K along with the eigen value of closed loop system E = *eig*(A-BK) can be obtained by MATLAB command [K, S, E] = *dlqr*(A, B, Q, R) for N = ∞.

For N = 8, the finite-time problem has solutions which has already reached constant optimal values for the present case. In general, however, the recursive method could only lead to the steady-state solutions by using as large a value of N as necessary. In this case since the

pair [A, B] is completely controllable, and we can find a 2 × 2 matrix D such that DD′ = Q and [A, D] is observable, the closed-loop system will be asymptotically stable for N = ∞. As a matter of fact the eigen values of the closed loop system are found to be

$$E = \lambda_{1,2} = [0.3746 + j\,0.2529 \quad 0.3746 - j0.2529]$$

which are inside the unity circle.

MATLAB SCRIPT

```
%Script_Example11.10
%Control sequence for finite interval discrete system
clear all; close all hidden
KK=[ ]
A1=[0 1;0 -4];b1=[0; 100];c1=[1 0];d1=[0];
g1=ss(A1,b1,c1,d1); g2=c2d(g1,0.1);
[A,B,C,D]=ssdata(g2)          % refer to equation (11.157)
R=[2];Q=[2 0;0 0.01];          % Equation (10.159)
[KTf,STf,e]=dlqr(A,B,Q,R)     % Infinite time value
j=8;x=[1;1];
warning off MATLAB:singularMatrix
S=[0 0;0 0]                    % Equation (11.163)
R_inv=inv(R+B'*S*B);  % part of Equ (11.147)
for j=7:-1:0;                  % Increments backward in time
j
K=R_inv*B'*S*A                 % refer to Equ (11.147)
S=Q+(A'*S)*(A-B*K)            % Ref Equation (11.148)
R_inv=inv(R+B'*S*B);          % part of Equ (11.147) changed recursively
KK=[KK,K'];                    % Feedback coefficients saved backward in time
end
i=0
x=[1; 1]; u=-KK(:,8)'*x
for i=1:8;
KKK=KK(:,9-i);                 % KKK arranged in forward time for
i                              % computing x and u
x=A*x+B*u
u=-KKK'*x                      % refer to Equ (11.146)
end
```

REVIEW EXERCISE

RE11.1 Consider the electric circuit shown in Fig. RE11.1 described by the equations

$$\dot{x}_1 = \frac{x_2}{L}$$

$$\dot{x}_2 = \frac{u - x_1}{C}$$

where L = inductance of the ideal inductor in Henry.
C = capacitance of the ideal condenser in Farad.
R = resistance in Ohm.

x_1 = current through the inductor in Ampere.

x_2 = voltage across the capacitor in Volts.

and u = input current in Ampere through the resistor R, assumed to be unconstrained with $x_1(0) = x_2(0) = 0$

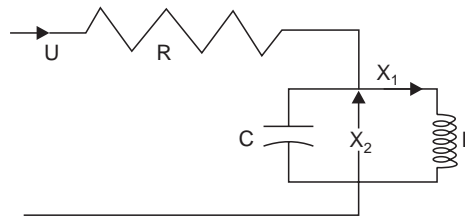



Fig. RE11.1

It is required to compute the optimal input current u so as to dissipate the least possible energy in the resistor R and bring the system, from rest [*i.e.* x_1 (0) = 0 and x_2 (0) = 0] at t = 0 to have at a given time T, voltage across the condenser x_2(T) =1 and zero current x_1 (T) = 0 flowing through the inductor.

Solution: Since dissipation in the resistor is to be minimized, the performance index may be written as:

$$J = \int_0^T R \frac{u^2}{2} dt, \text{ with the final set } S_f \text{ is simply ([0 1], T).}$$

Taking the Hamiltonian function as $H = R\frac{u^2}{2} + p_1 \frac{x_2}{L} + p_2 \frac{u - x_1}{C}$

where, for simplicity, T may be taken as : $T = \pi/\omega$, $\omega = 1/\sqrt{LC}$.

Using Equation (11.91), we get $u^0 = -\frac{p_2}{CR}$. Therefore, the optimal value of H is given by :

$$H^0(x, p, t) = -\frac{1}{2}\frac{p_2^2}{C^2R} + p_1 \frac{x_2}{L} - p_2 \frac{x_1}{C}$$

So, using (11.89) the state equation becomes : $\dot{x}_1 = \frac{x_2}{L}$ and $\dot{x}_2 = -\frac{x_1}{C} - \frac{p_2}{C^2R}$

and from Equation (11.90) we get $\dot{p}_1 = -\frac{p_2}{C}$, $\dot{p}_2 = \frac{p_1}{L}$

From the last two equations we can easily solve for p_2 as $p_2(t) = k_1 \cos \omega t$, where $w = 1/\sqrt{LC}$, and k_1 is a constant.

Putting this value of $p_2(t)$ in the differential equation, $p_1(t)$ is found as : $p_1(t) = -L\,\omega\, k_1 \sin(\omega t)$.

The state equation can now be solved using the value of $p_2(t)$. The solution for $x_1(t)$ is given by:

$$x_1(t) = -\frac{k_1}{C^2\, LR} t \sin(\omega t)$$

Putting this value of $x_1(t)$ in the state equation we have

$$x_2(t) = L\dot{x}_1 = -\frac{k_1}{C^2R} [\sin(\omega t) + (\omega t) \cos(\omega t)]$$

The constant k_1 is found by noting the boundary condition at $t_f = T = \pi/\omega$, $x_2(T) = 1$, which yields $k_1 = \dfrac{C^2R}{\pi}$. The boundary condition on $x_1(t)$ are also satisfied: at $t = 0$, $x_1(0)= 0$ and at $t = T$, $x_1(T) = 0$.

Substituting the value of k_1 the optimal control is found as $u^0(t) = -\dfrac{C}{\pi}\cos(\omega t)$

which causes the state response: $x_1^{\,0}(t) = -\dfrac{1}{L\pi}\, t \sin(\omega t)$

$$x_2^{\,0}(t) = -\frac{1}{\pi}\,[\sin(\omega t) + \omega t \cos(\omega t)],$$

PROBLEMS

P11.1 For the system $\dot{x} = -4x + u$ find the optimal control $u(t)$ which drives the system from the initial state $x(0) = 1$ to the terminal state $x(1) = 0$, so as to minimize the performance index $J = \dfrac{1}{2}\displaystyle\int_0^1 u^2\, dt$. Also find the optimal state trajectory.

Ans: $u^o(t) = \dfrac{-8e^{4t}}{e^8 - 1}$ and $x^o(t) = e^{-4t} - \dfrac{e^{4t} - e^{-4t}}{e^8 - 1}$

P11.2 Find the optimal control $u(t)$ for the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u$ that will transfer the initial state $x'(0) = [1\ 1]$ to the final state $x'(1) = [0\ 0]$, and minimize the performance index $J = \dfrac{1}{2}\displaystyle\int_0^1 u^2\, dt$. Also find the optimal state trajectory.

Ans: $u(t) = 9t - 5$; $x_1(t) = 1 + 3t^3 - 5t^2 + t$ and $x_2(t) = 1 + 9t^2 - 10t$

P11.3 Find the optimal control for the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & 0 \end{bmatrix} x + \begin{bmatrix} 0 \\ 4 \end{bmatrix} u$ with initial condition $x'(0) = [1\ 1]$ and terminal condition $x'(1) = [0\ 0]$ so as to minimize the performance index $J = \dfrac{1}{2}\displaystyle\int_0^1 u^2\, dt$.

Ans: $u(t) = a_1 \sin(2t) - a_2 \cos(2t)$, where $a_1 = [4 + 2\tan(2) + 5\tan^2(2)]/[4 + 3\tan^2(2)] = 1.2826$ and $a_2 = [2 - \tan(2) + 4\tan^2(2)]/[4 + 3\tan^2(2)] = 1.2707$

P11.4 For the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u$ assuming unconstrained u, find the expressions for $\dot{x}$ and $\dot{p}$ in terms of the Hamiltonian function from equations (11.90) and (11.91) with respect to the performance indices (*a*) $J = \dfrac{1}{2}\displaystyle\int_0^1 (x_1^2 + u^2)\, dt$ and (*b*) $J = \dfrac{1}{2}\displaystyle\int_0^1 (x_1^2 + x_2^2 + u^2)\, dt$

Hints: (*a*) Here Hamiltonian function is given by : $H = x_1^2 + u^2 + p_1x_2 + p_2(-4x_1 - 5x_2 + 2u)$ Using relation (11.91), find u^o as $u^o = -p_2$, then use the relations (11.89) and (11.90) to get the result.

Ans: (*a*)

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix} p$$

$$\dot{p} = \begin{bmatrix} -2 & 0 \\ 0 & 0 \end{bmatrix} x + \begin{bmatrix} 0 & 4 \\ -1 & 5 \end{bmatrix} p$$

and (*b*) $\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} x + \begin{bmatrix} 0 & 0 \\ 0 & -2 \end{bmatrix} p$

$\dot{p} = \begin{bmatrix} -2 & 0 \\ 0 & -2 \end{bmatrix} x + \begin{bmatrix} 0 & 4 \\ -1 & 5 \end{bmatrix} p$

P11.5 Find the optimal-feedback coefficients for the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -3 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$ if the performance index is $J = \int_0^\infty (x_1^2 + x_2^2 + ru^2)\, dt$ where (*a*) $r = 0.2$ (*b*) $r = 1.0$ and (*c*) $r = 5.0$. Compare the response of the three systems for an initial condition of $x(0) = 1$.

Ans: (*a*) k = [1.0000 1.0000] (*b*) [0.2361 0.2361] and (*c*) [0.0494 0.0494]

P11.6 Find the optimal-feedback coefficients for the system

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -2 & -1 \end{bmatrix} x + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

if the performance index is $J = \int_0^\infty (x_1^2 + u^2)\, dt$.

Ans: k' = [0.2361 0.2133]

P11.7 Find the optimal-feedback coefficients for the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ -4 & -5 \end{bmatrix} x + \begin{bmatrix} 0 \\ 2 \end{bmatrix} u$ if the performance index is $J = \int_0^\infty (\alpha x_1^2 + u^2)\, dt$ where (*a*) $\alpha = 1.0$ (*b*) $\alpha = 5.0$

Ans: (*a*) k' = [0.2361 0.0468] and (*b*) k' = [1.0000 0.1926]

P11.8 Solve problem 11.7 changing the performance index to $J = \int_0^\infty (x_1^2 + x_2^2 + u^2)\, dt$

Ans: k' = [0.2361 0.2361]

P11.9 Find the optimal-feedback-coefficient matrix for the system $\dot{x} = \begin{bmatrix} 0 & 1 \\ 2 & 2 \end{bmatrix} x + \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} u$ if the performance index is $J = \int_0^\infty (x_1^2 + x_2^2 + u_1^2 + u_2^2)\, dt$

Ans: $k' = \begin{bmatrix} 1.0000 & 1.0000 \\ 2.0000 & 3.4142 \end{bmatrix}$

P11.10 For the system $\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -4 & -5 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u$ and the performance index

$$J = \int_0^\infty (x_1^2 + x_2^2 + 2u_1^2 + u_2^2)\, dt$$

Ans : $K' = \begin{bmatrix} 0.6971 & 0.4828 & 0.0839 \\ 0.1678 & 0.2971 & 0.0577 \end{bmatrix}$

P11.11 For the system $\dot{x} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -4 & -5 \end{bmatrix} x + \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{bmatrix} u$ if the performance index is

$$J = \int_0^\infty (x_1^2 + x_2^2 + u_1^2 + 2u_2^2)\, dt$$

Ans: $K' = \begin{bmatrix} 0.9976 & 0.5935 & 0.0986 \\ 0.0493 & 0.1131 & 0.0221 \end{bmatrix}$

CHAPTER

12

Fuzzy Logic for Control System

12.1 THE CONCEPT OF FUZZY LOGIC AND RELEVANCE OF FUZZY CONTROL

In order to design a controller for a system a mathematical model of the plant or system is needed. However, for some control systems, preparing a mathematical model is extremely difficult, if not impossible. Consider, for example, preparing a mathematical model of the system for the following tasks:

(*a*) Parking a car in a parking lot

(*b*) Controlling traffic at road intersections

(*c*) Washing clothes

(*d*) Controlling a cement kiln

(*e*) Waste effluent treatment in a plant

(*f*) Judging beauty contestants

The difficulty generally arises from uncertainty in the form of ambiguity in information and data. However, expert drivers, operators and judges are doing the above tasks routinely. The complete description of a real world system often requires more detailed data than a human could ever hope to handle them properly and efficiently. They do it by reasoning approximately about the system-behavior, thereby maintaining only a generic understanding about the problem. Fortunately, this generality and ambiguity are sufficient for human reasoning about the complex systems in most of the situations. One need not be very fastidious about accuracy when a rough answer will serve the purpose. Fuzzy logic is all about this relative importance of precision in presence of complexity and ambiguity (imprecision). According to Zadeh [39], *the closer one looks at a real-world problem, the fuzzier becomes its solution.*

A closed-form mathematical expression can provide precise descriptions of a system with little complexity (and hence little uncertainty). But for the most complex systems where few numerical data exist and where only ambiguous or imprecise information may be available, fuzzy reasoning can provide a way to understand system behavior by allowing us to interpolate approximately between observed input and output situations.

Natural language is our own means of communication: By its very nature, natural language is vague and imprecise; yet it is the most powerful form of communication and information exchange among humans. Despite the vagueness in natural language, humans have little trouble understanding one another's concepts and ideas; this understanding is not possible in communications with a computer, which requires extreme precision in its instructions. For instance, what is the meaning of a *tall person* ? To individual X, a tall person might be anybody over 5′11". To individual Y, a tall person is someone who is 6′2" or taller. What sort of meaning does the linguistic descriptor *tall* convey to either of these individuals?

It is surprising that, despite the potential for misunderstanding, the term *tall* conveys sufficiently similar information to the two individuals, even if they are significantly different heights themselves, and that understanding and correct communication are possible between them. Individuals X and Y, regardless of their own heights, do not require identical definitions of the term *tall* to communicate effectively; again, a computer would require a specific height to compare with a pre-assigned value for *tall*. The underlying power of fuzzy set theory is that it uses *linguistic* variables, rather than *quantitative* variables, to represent imprecise concepts.

The incorporation of fuzzy set theory and fuzzy logic into computer models has shown tremendous payoff in areas where intuition and judgment still play major roles in the model. Control applications, such as temperature control, traffic control, or process control, are the most prevalent of current fuzzy logic applications. Fuzzy logic seems to be most successful in two kinds of situations: (*i*) very complex models where understanding is strictly limited or, in fact, quite judgmental, and (*ii*) processes where human reasoning, human perception, or human decision making are inextricably involved. Generally, simple linear systems or naturally automated processes have not been improved by the implementation of fuzzy logic.

Our understanding of physical processes is based largely on imprecise human reasoning. This is nonetheless a form of information that can be quite useful to humans. The ability to embed such reasoning in hitherto intractable and complex problems is the criterion by which the efficacy of fuzzy logic is judged. However, this ability cannot solve problems that require precision-problems such as shooting precision laser beams over tens of kilometers in space; milling machine components to accuracy's of 1 parts per million; or focusing a microscopic electron beam on a specimen the size of a nanometer. But not many tasks require such precision-tasks such as parking a car, backing up a trailer, navigating a car among others on a highway, washing clothes, controlling traffic at intersections, judging beauty contestants, and so on.

However, for most systems, higher precision entails higher cost. When considering the use of fuzzy logic for a given problem, one should ponder the need for *exploiting the tolerance for imprecision*. In this context, consider the *traveling sales representative problem* mentioned in footnote in Chapter 1. There are real, practical problems analogous to the traveling sales representative problem. Such problems arise in the fabrication of circuit boards, where precise lasers drill hundreds of thousands of holes in the board. Deciding in which order to drill the holes (where the board moves under a stationary laser) to minimize drilling time is a traveling sales representative problem [6].

Some wise men have given their opinion about the relative importance of precision in a complex situation that we encounter in our daily life quite often. Here we quote:

Sometimes the more measurable drives out the most important. —Reno Dubos

Vagueness is no more to be done away with in the world of logic than
Friction in mechanics (and resistive loss in electric circuits) —Charles Sanders Peirce

So far as the laws of mathematics refer to reality, they are not certain.
And so far as they are certain, they do not refer to reality. —Albert Einstein

As complexity rises, precise statements lose meaning and meaningful
statements lose precision. —Lotfi Zadeh

Some pearls of folk wisdom also echo these thoughts:

Don't lose sight of the forest for the trees.

Don't be penny-wise and pound-foolish.

Fuzzy logic is a fascinating area of current research since it is doing a good job of trade off between significance and precision–something that humans have been managing for a very long time.

The concepts of fuzzy logic are as old as human history, but its systematic study has begun during the last four decades.

12.2 INDUSTRIAL AND COMMERCIAL USE OF FUZZY LOGIC-BASED SYSTEMS

In the decade after Dr. Zadeh's seminal paper on fuzzy sets [2], many theoretical developments in fuzzy logic took place in the United States, Europe, and Japan. From the mid-Seventies to the present, however, Japanese researchers have been a primary force in advancing the practical implementation of the theory; they have done an excellent job of commercializing this technology.

Fuzzy logic affects many disciplines. In videography, for instance, Fisher, Sanyo, and others make fuzzy logic camcorders, which offer fuzzy focusing and image stabilization. Mitsubishi manufactures a fuzzy air conditioner that controls temperature changes according to human comfort indexes. Matsushita builds a fuzzy washing machine that combines smart sensors with fuzzy logic. The sensors detect the color and kind of clothes present and the quantity of grit, and a fuzzy microprocessor selects the most appropriate combination from 600 available combinations of water temperature, detergent amount, and wash and spin cycle times. The Japanese City of Sendai has a 16-station subway system that is controlled by a fuzzy computer. The ride is so smooth that the riders do not need to hold straps, and the controller makes 70 percent fewer judgmental errors in acceleration and braking than human operators do. Nissan introduced a fuzzy automatic transmission and a fuzzy anti-skid braking system in one of their recent luxury cars. Tokyo's stock market has stock-trading portfolios based on fuzzy logic that outperformed the Nikkei Exchange average. In Japan, there are fuzzy golf diagnostic systems, fuzzy toasters, fuzzy rice cookers, fuzzy vacuum cleaners, and many other industrial fuzzy control processes.

12.3 FUZZY MODELING AND CONTROL

Fuzzy control may be looked upon as a real-time expert system, implementing a part of a human operator's or process engineer's expertise which does not lend itself to being easily expressed in differential equations, but rather in situation/action rules. However, fuzzy control differs from mainstream expert system technology in many ways. One of its striking features is that it exists at two distinct levels. It can be described in (*a*) symbolic if-then rules as well as in (*b*) linguistic variables and linguistic values such as:

if *pressure is high and slightly increasing* then *steam flow is medium negative*

The linguistic values such as *slightly increasing* and fuzzy operators such as *and* may be compiled into elementary numerical objects and algorithms like function tables, interpolation and comparators. The compiled information may be incorporated in a real-time implementations of fuzzy control by embedding into the numerical environment of conventional control.

From this perspective, fuzzy control can be seen as a heuristic and modular way for defining nonlinear, table-based control systems. The rule above may be considered an informal "nonlinear PD-element". A collection of such rules will provide the definition of a nonlinear transition function, without the need for defining each entry of the table individually, and without the knowledge of closed form representation of that function. One way to combine fuzzy and PD-control then is to use a linear PID-system around the set point, where it does its job, and let the fuzzy controller to take over when the variables are far off from the quiescent operating point.

12.3.1 Advantages of Fuzzy Controller

Considering the existing applications of fuzzy control, which range from very small, micro-controller based systems in home appliances to large-scale process control systems, the advantages of using fuzzy control usually fall into one of the following four categories:

(*a*) Fuzzy Control provides higher level of automation by incorporating expert knowledge. In many cases of process control as in chemical industries, the level of automation is quite low. There is a variety of basic, conventional control loops, but a human operator is needed during the starting or shutting down phase, for tuning controllers, or for switching between different control modules. The knowledge of the operator is usually based on experience and does not lend itself to being expressed in differential equations. It is often rather of the type *if the situation is such and such, one should do the following*. In this case, fuzzy control offers a method for representing and implementing the expert's knowledge. As an example, a Portuguese paper manufacturing company has attained a reduction up to 60% of the variation of product quality by incorporating the core of the operator's control strategy in just twenty-five rules in fuzzy-based digester management system on top of its process control software. Furthermore, subsequent optimizations of the software system led to a significant reduction of energy and base material consumption. A return on the investment made on the fuzzy control software package and the knowledge base of the actual control system, was achieved within few months.

(*b*) Fuzzy controller provides Robust nonlinear control. Consider the problem of controlling a robot arm with several links used to move objects with different masses along a predefined path. Since there are good and exact models of this system available, it is a simple job to realize a PID-controller that works pretty well for known masses within a narrow range. However, in presence of large parameter variations or major external disturbances the PID controllers are usually faced with a trade-off between (*i*) fast response with significant overshot or (*ii*) smooth but slow response or (*iii*) even resulting in a system that is not stable at all. In this case, fuzzy control offers ways to implement simple but robust solutions that cover a wide range of system parameters and that can cope with major disturbances. In this particular case, a so-called *fuzzy sliding mode controller* can be implemented, that exhibits similar performance for a given mass with only slight variations, but outperforms the PID controller in presence of large disturbances.

(*c*) Fuzzy controller needs reduced development and maintenance time. In many cases, two different groups of experts—(*i*) the process and field experts and (*ii*) electronics and systems programming engineers, are involved in the development of a control system. Process experts know the application problem and control strategy, but may not be experts of numerical algorithms and micro-controllers. The actual system realization is carried out by electronics and systems programming engineers, who on the other hand may not be familiar with the application problem. This often increases the communication gap between these two groups of people, resulting in delays and extension of development time. Fuzzy control, which 'inhabits' at two levels of abstraction, offers languages at both levels of expertise—the *symbolic level* is appropriate for describing the application engineers' strategies while the *compiled level* is well understood by the electronic engineers. Since there is a well-defined, formal translation between those levels, a fuzzy based approach can help reduce communication gap and hence the development time and cost.

As an illustration, consider controlling idle-speed in automotive electronics, where sophisticated control strategies have to be implemented and run in time-critical situations on simple 8-bit micro-controllers. The major control goal was to keep a constant idle-speed of 800 rpm, irrespective of disturbances imposed by road surface conditions or additional loading

such as power steering and air conditioning. It took almost two man-years to develop a conventional PID-system for accommodating major variations in system parameters in mass produced cars and aging problems that needs extensive system tuning. The problem of communication of engine experts about their control strategy to the micro controller specialists had its share in the extended completion time. The development time for the fuzzy controller was around only six months and practically no differences could be observed in the system's performance.

There is yet another reason, however, for using fuzzy control even if it does not improve system performance or reduce development costs. In some business areas, the patent situation is such that it is hard to come up with new solutions using conventional technology without violating or at least interfering with a competitor's patent. In some cases, using fuzzy control for a qualitatively equivalent solution will and actually does help to by-pass existing patents.

12.3.2 When to Use Fuzzy Control

There should not be too high expectations from fuzzy controllers after seeing a few successful applications, since it is not a panacea of all control problems. Naturally, the question then arises when the fuzzy control should be considered ?

A fully satisfactory answer to this question is yet to be formulated. Since the industrial and commercial use of fuzzy control has started only from late eighties, compared to the numerous existing conventional control systems with an accumulated knowledge spread over many decades, the statistical support in terms of the number of successful applications is not broad enough for any generalization. So the best option, at this point, is to provide a few guidelines on whether to use fuzzy control or not in a given situation.

(1) If there already exists a good PID controller where system performance, development and maintenance costs as well as marketing policy are satisfactory, it should be retained.

(2) If there exists a successful fuzzy controller to a problem similar to the one at hand, the chance is that fuzzy controller may click for the present case too.

(3) Finally, if the existing solution with respect to any one of these criteria are not satisfactory, or if one has to deal with a problem that remained unsolved so far, the reason should be analyzed. If adequate knowledge is available about the system or process that could be used to improve the solution but that is hard to encode in terms of conventional control such as differential equations and PID-parameters, fuzzy control may be tried. In many cases, this knowledge is qualitative and heuristic in nature and amenable to fuzzy if-then rules.

12.3.3 Potential Areas of Fuzzy Control

With increasing complexities in system engineering, the focus of fuzzy control is moving from elementary control problems to higher levels in the system hierarchy such as supervisory control, monitoring and diagnosis, and logistic support. It is to be noted that telecommunications, which is one of the major future industries, has started investigating fuzzy control for communication systems and that several pilot projects have been initiated for tackling routing and overload handling problems.

So far, the majority of existing applications are purely software-based. However, general-purpose fuzzy logic processors or coprocessors will be found to be useful in extremely time-critical applications like pattern recognition task in a complex plant automation and in mass-produced automotive electronics.

The *first generation* of fuzzy control in the existing applications exploits only a very small fragment of fuzzy logic theory. In many cases of more complex, ill-structured problems,

this first generation technology is not sufficiently equipped to represent and implement the knowledge needed for powerful solutions.

Besides, there is strong need for a more systematic design and analysis methodology for fuzzy control applications, spanning the whole life-cycle from perception to all the way up to deployment and maintenance. It must provide answers to make a proper choice of alternative design issues after a thorough analysis of the problem, and must be able to associate variations of parameters to system-performance. At this stage, one should not expect a universal design and optimization strategy for fuzzy control, which will be of some practical use. Such a universal theory does not exist for conventional control engineering either. Instead, we have to proceed from the few isolated islands where we already know exactly how to design a fuzzy control algorithm to clusters of problems and related design methodologies.

From the above discussions it is apparent that fuzzy control has tremendous scope in the knowledge based systems approach to closed loop control system, which may be defined as:

A knowledge based system for closed loop control is a system which enhances the performance, reliability and robustness of control by incorporating knowledge which can not be captured in the analytical model used for controller design and that is taken care of by manual modes of operation or by other safety and ancillary logic mechanism.

12.3.4 Summary of Some Benefits of Fuzzy Logic and Fuzzy Logic Based Control System

We sum up below the strengths of fuzzy logic and Fuzzy Logic based control even though we must have patience as the discipline matures to attain its full potential.

1. **Fuzzy logic is conceptually easy to understand.** The mathematical concepts behind fuzzy reasoning are very simple and natural.
2. **Fuzzy logic is flexible.** With any given system, one can add incrementally more functionality to the solution without restarting from scratch.
3. **Fuzzy logic is tolerant of imprecise data.** Most things are imprecise on closer inspection. Fuzzy reasoning is adept in incorporating this imprecision into the process model at the very outset and does not wait till the fag end of design-stage for its inclusion.
4. **Fuzzy logic can model nonlinear functions of arbitrary complexity.** One can create a fuzzy system to map any set of input data to output with arbitrary precision. This is exploited by adaptive techniques like Adaptive Neuro-Fuzzy Inference Systems.
5. **Fuzzy logic can be built on top of the experience of experts.** Fuzzy IF Then rules can readily capture the expert knowledge which is hard to model in any other format including differential equations.
6. **Fuzzy logic can be blended with conventional control techniques.** Fuzzy controller can be successfully added to play a complementary role to an existing conventional controller and simplify its implementation.
7. **Fuzzy logic is based on natural language.** The basis for fuzzy logic is the basis for human communication. This observation underpins many of the other statements about fuzzy logic.

Natural language has been shaped by thousands of years of human history to be convenient and efficient. Sentences written in ordinary language represent a triumph of efficient communication. We are generally unaware of this because we use natural language every day. Since fuzzy logic is built atop the structures of qualitative description used in everyday language, fuzzy logic is easy to use.

12.3.5 When Not to Use Fuzzy Logic

Fuzzy controller is not a panacea to all control problems we may encounter. It is true that in many cases the fuzzy control leads to a higher degree of automation for complex and ill-structured processes, provided relevant knowledge about the process together with the control strategy is available, that can be well expressed in terms of fuzzy logic. There are processes for which that kind of knowledge is either simply not available or not adequate.

Fuzzy logic is a convenient way to map an input space to an output space. If you find it's not convenient, try something else. If a simpler solution already exists, use it. Many controllers, for example, do a fine job without using fuzzy logic. However, if you take the time to become familiar with fuzzy logic, you'll see it can be a very powerful tool for dealing quickly and efficiently with imprecision and nonlinearly.

At present, there is a worldwide shortage of well-trained personnel in fuzzy control, which is preventing a broader exploitation of this new technology. However, the situation is likely to change soon when the new crop of well trained personnel enters the industrial scene from the universities and training institutes, who have taken keen interest in integrating Fuzzy control into the academic curricula.

12.4 FUZZY SETS AND MEMBERSHIP

In his seminal paper of 1965, Dr Lotfi Zadeh [2] suggested that *set membership* is the key to decision making when faced with uncertainty.

> *The notion of a fuzzy set provides a convenient point of departure for the construction of a conceptual framework which parallels in many respects the framework used in the case of ordinary sets, but is more general than the latter and, potentially, may prove to have a much wider scope of applicability, particularly in the fields of pattern classification and information processing. Essentially, such a framework provides a natural way of dealing with problems in which the source of imprecision is the absence of sharply defined criteria of class membership rather than the presence of random variables.*

The decision whether someone is over 6 feet tall is a easy one. In a binary world of Aristotelian logic, the person either is or is not, based on the accuracy, or imprecision, of our measuring device. If we define a set of people *tall* whose heights are equal to or greater than 6 feet, a computer would not recognize an individual with height 5′11.999" as being a member of the set "tall." But how do we deal with the uncertainty in the following statement: The person is *nearly* 6 feet tall. The uncertainty in this case is due to the vagueness or ambiguity of the adjective *nearly.* A 5′11″ person could clearly be a member of the set of "nearly 6-feet tall" people. In the first situation, the uncertainty of whether an unknown person is 6 feet or not is binary; the person either is or is not, and one can produce a probability assessment of that prospect based on height data from many people. But the uncertainty of whether a person is nearly 6 feet is nonrandom. The degree to which the person approaches a height of 6 feet is fuzzy. In reality, *tallness* is a matter of degree and is relative. So, 6 feet can be tall in one context and short in another. In the real ("gray" or fuzzy) world, the *set of tall people* can overlap with the *set of not-tall people*, which is an impossibility in the world of binary logic. This notion of set membership defined on the universe, then, is central to the representation of objects within a universe.

12.4.1 Introduction to Sets

In this section we introduce the some basic definitions, notation, and operations for fuzzy sets that will be needed in the discussions that follow.

Let X represent a space of objects (called the universe *of discourse,* or universal *set)* which contains all the possible elements of our interest in a given context and x is its generic element. We can define a *classical (crisp) set* A, A $\in$ X (or simply a set A) in the universe of discourse X by listing all of its members (*the list method*) or by specifying the properties that the elements of the set must satisfy (*the rule method*). The rule method is more general compared to the list method, which is applicable only to sets with finite elements. In a rule method a set A is represented as

$$A = \{x \in X \mid x \text{ satisfies some conditions}\} \tag{12.1}$$

There is a third method to define a set A, known as *the membership method*, in which a membership function (also called characteristic or indicator function) is associated with each element of the set A. The characteristic function or membership function of an element x in the set A is denoted by $\mu_A(x)$ and is written as

$$\mu_A(x) = \begin{cases} 1 & \text{for } x \in A \\ 0 & \text{for } x \notin A \end{cases} \tag{12.2}$$

We can, therefore, represent a classical set A by a set of ordered pairs $(x, 1)$ or $(x, 0)$, which indicates x $\in$ A or $x \notin$ A, respectively.

$$A = \{(x, 1) \mid x \in X\}$$

$$A = \{(x, 0) \mid x \notin X\} \tag{12.3}$$

12.4.2 Classical Sets

A classical set is a set with a crisp boundary. For instance, a classical set A of real numbers greater than 6 can be represented as

$$A = \{x \mid x > 6\} \tag{12.4}$$

where the unambiguous boundary is at 6 such that if x is greater than 6, then x belongs to the set A; otherwise x does not belong to the set A. Classical sets have played an important role in mathematics and computer science, but they failed to capture the entire gamut of human thoughts, which is abstract and imprecise. For example, we can have a collection of *tall* persons whose height is more than 6 ft by substituting A = "tall person" and x = "height." in Equation (12.4).

Suppose a particular individual, x_1 has a height of 6.001 feet. The membership of this individual in crisp set A is equal to 1, or full membership, written symbolically as $\mu_A(x_1) = 1$. Another individual, say, x_2, has a height of 5.999 feet. The membership of this individual in set A is equal to 0, or no membership, hence $\mu_A(x_2) = 0$, (vide Figure 12.1(*a*)). In these cases the membership in a set is binary, either an element is a member of a set or it is not.

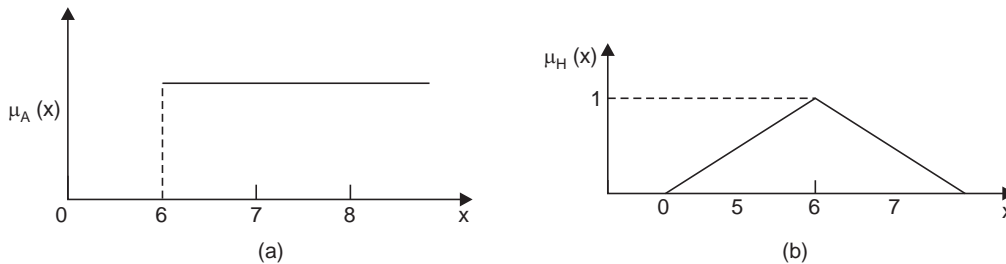



Fig 12.1 Height membership function for (*a*) crisp set A and (*b*) a fuzzy set H

But this is against our normal concept of "tall person" whereby a person with a height of 6.001 ft is tall, and another with a height of 5.999 ft is not. This distinction is intuitively

unreasonable. The flaw comes from the sharp transition between inclusion and exclusion in a set.

12.4.3 Fuzzy Sets

The notion of binary membership can be extended to accommodate various "degrees of membership" on the real continuous interval [0, 1], where the end points of 0 and 1 conform to no membership and full membership, respectively, just as the indicator function does for crisp sets. But the infinite number of values in between the endpoints 0 and 1 can represent various degrees of membership for an element x in some set on the universe. The sets on the universe X that can accommodate "degrees of membership" were termed by Zadeh as "fuzzy sets." Fuzzy set, therefore, is a set without a crisp boundary. That is, the transition from "belong to a set" to "not belong to a set" is gradual, and this smooth transition is characterized by membership functions that give fuzzy sets flexibility in modeling commonly used linguistic expressions, such as "the milk is hot" or "the speed is high." As Zadeh pointed out in 1965 in his seminal paper entitled "Fuzzy Sets" [2], such imprecisely defined sets or classes play an important role in human thinking, particularly in the domains of pattern recognition, communication of information, and abstraction". Note that the fuzziness does not come from the randomness of the constituent elements of the sets, but from the uncertain and imprecise nature of abstract thoughts and concepts. A fuzzy set expresses the degree to which an element belongs to a set. Hence the characteristic function of a fuzzy set is allowed to have values between 0 and 1, which denotes the degree of membership of an element in a given set.

Continuing further on the example on heights, consider a set H consisting of heights *around 6 feet.* Since the property *around 6 feet* is fuzzy, there is not a unique membership function for H. Rather, one must decide, in a given situation, about the nature of the membership function, denoted by μ_H. Some of the properties of this function are : (*i*) normality ($\mu_H(6) = 1$), (*ii*) monotonicity (the closer H is to 6, the closer μ_H to 1), and (iii) symmetry (numbers equidistant from 6 should have the same value μ_H). Such a membership function is illustrated in Figure 12.1(*b*). A key difference between crisp and fuzzy sets is their membership function; a crisp set has a unique membership function, whereas a fuzzy set can have an infinite number of membership functions to represent it. For fuzzy sets, the uniqueness is sacrificed, but flexibility is gained because the membership function can be adjusted to maximize the utility for a particular application.

Let us now set forth several basic definitions concerning fuzzy sets.

12.5 BASIC DEFINITIONS OF FUZZY SETS AND A FEW TERMINOLOGIES

Definition 12.1 *Fuzzy sets and membership functions*

If X is a collection of objects or elements x, then a **fuzzy set** A in X is defined as a set of ordered pairs:

$$A = \{(x, \mu_A(x)) \mid x \in X\} \tag{12.5}$$

where $\mu_A(x)$ is called the **membership function** (**MF**) for the fuzzy set A. The MF functions maps or associates each element x of X to a real number in the interval [0, 1]. The value of the membership function at element x represents the "grade of membership" .

So, the definition of a fuzzy set is a simple extension of the definition of a classical set in which the indicator function is permitted to have any values between 0 and 1. If the value of the membership function $\mu_A(x)$ is restricted to take on discrete values of 0 or 1, then A is reduced to a classical set and $\mu_A(x)$ is the indicator function of discrete set A. We shall refer to classical sets as crisp sets, or just sets to avoid any confusion.

Membership grades are fixed only by convention and the unit interval as a range of membership grades, is arbitrary.

The unit interval is natural for modeling membership grades of fuzzy sets of real numbers.

The **universe of discourse** X may consist of discrete (ordered or non-ordered) objects or continuous space. This is illustrated with the following examples.

Example 12.1 *Fuzzy sets with a discrete non-ordered universe*

Let X = {Calcutta, Delhi, Bangalore} be the set of cities where one may want to build a home. The fuzzy set

C = "desirable city to build a home" may be described as follows:

C = {(Calcutta, 0.8), (Delhi, 0.9), (Bangalore, 0.6)}.

The universe of discourse X, here is obviously discrete and it contains non-ordered objects like three big cities in India. Obviously, the assignment of membership grades to the cities are quite subjective and anyone is free to come up with three different values in [0 ,1] to indicate his or her preference.

Example 12.2 *Fuzzy sets with discrete ordered universe*

Let U be the number of students from 1 to 10 , U={1,2,3, … 9,10}. Then, using summation notation (vide Equation 12.7 below), the fuzzy set *several students* may be defined as

Several students = 0/1+0.2/2+0.5/3+0.8/4+1.0/5+1.0/6+0.8/7+0.5/8+0.2/9+0/10

That is 5 and 6 students belong to the fuzzy set of "several students" with membership grade 1, 4 and 7 students with grade 0.8, 3 and 8 students with grade 0.5 and 2 and 9 students with grade 0.2 (vide Figure 12.2*a*).

Example 12.3 *Fuzzy sets with a continuous universe*

Let X = R^+ be the set of possible ages for people. Then the fuzzy set *B* = "about 25 years old" may be expressed as

$$B = \{(x, \mu_B(x)) \mid x \in X\}, \quad \text{where } \mu_B(x) = 1/\{(1 + 0.2(x - 25)^4\} \tag{12.6}$$

This is illustrated in Figure 12.2(*b*).

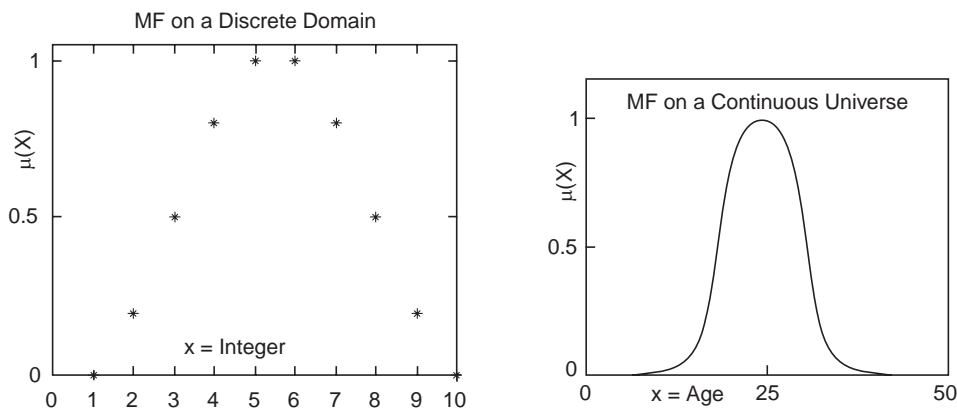



Fig 12.2 (*a*) Membership function for "several students" and (*b*) "about 25 years old"

It is obvious, therefore, that we need two things to construct a fuzzy set: identification of a suitable universe of discourse X and selection of a suitable membership function. The selection of membership functions is highly subjective. Therefore, one may use different membership

functions to characterize the same description. However, the membership functions themselves are not fuzzy, they are precise mathematical functions. Once a fuzzy property is represented by a membership function, for example, once "about 25 years old" is represented by the membership function (12.6), nothing will be fuzzy anymore. Thus, by characterizing a fuzzy description with a membership function, we essentially defuzzify the fuzzy description. That is, when we say a fuzzy set, there must be a unique membership function associated with it; conversely, when we give a membership function, it represents a fuzzy set. Fuzzy sets and their membership functions are equivalent in this sense.

The subjectivity in the membership function comes from individual differences in perceiving abstract concepts and has little to do with randomness. Therefore, the **subjectivity** and **non-randomness** of fuzzy sets is the primary difference between the study of fuzzy sets and probability theory, which deals with objective treatment of random phenomena.

For convenience, an alternative way of denoting a fuzzy set is introduced below. A fuzzy set A can be written as:

$$A = \begin{cases} \Sigma_{x_i \in X} \mu_A(x_i)/x_i & \text{if X is a collection of discrete objects} \\ \int X \mu_A(x_i)/x_i & \text{if X is a continous space (Real line } R^+) \end{cases} \tag{12.7}$$

The summation and integration signs in Equation (12.7) stand for the union of pairs $(x, \mu_A(x))$; they have nothing to do with summation or integration. Similarly, "/" is only a marker and does not indicate division.

Example 12.4. *Membership grades expressed in alternative form*

The fuzzy sets in Examples 12.1, 12.2, and 12.3 when expressed in alternative notation of Equation (12.7) appear as :

C = 0.9/Calcutta + 0.8/Delhi + 0.6/Bangalore,

A = 0.0/1+0.2/2+0.4/3+0.8/4+1.0/5+1.0/6+0.8/7+0.4/8+0.2/9+0.0/10

and $B = \int_{R^+} \frac{1}{1+0.2\,(x-25)^4} \Big/ x$ respectively.

The continuous space of discourse X (the real line R^+ or its subset), is partitioned, for convenience in a practical situation, into several fuzzy sets to cover the range [0, 1] and identified with suitable terms familiar to humans like "large, "medium," or "small,". These terms are called linguistic values or linguistic labels. Under these circumstances, the universe of discourse X may be called the linguistic space and its elements are called linguistic variables [1].

Example 12.5. *Linguistic variables and linguistic values*

If the linguistic variable "age" is represented by X, then we can define fuzzy sets "young," "middle aged," and "old" that are characterized by MFs $\mu_{young}(x)$, $\mu_{middle\text{-}aged}(x)$, and $\mu_{old}(x)$, respectively. A linguistic variable "Age" can also assume different linguistic values, such as "young," "middle aged," and "old" in the same way an algebraic variable can assume any values assigned to it. If the "age" of a person is represented by the value "young," then we call the person " young" and if the "age" of a person takes the value "old," then we call the person "old". Figure 12.3 shows typical MFs for these linguistic values. The universe of discourse X is totally covered by the MFs and the transition from one MF to another is smooth.

12.5.1 Commonly Used Fuzzy Set Terminologies

A fuzzy set is uniquely specified by its membership function in the universe of the real line R. We shall define the nomenclatures used in the literature for more specific description of membership functions.

Definition 12.2 *Support* The **support** of a fuzzy set A is the set of all points x in X such that $\mu_A(x) > 0$:

$$\text{support (A)} = \{x \mid \mu_A(x) > 0\} \tag{12.8}$$

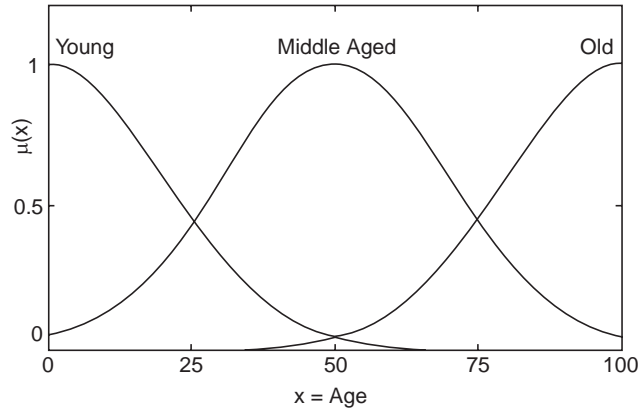



Fig 12.3 Typical membership values of "young", "middle aged" and "old"

Definition 12.3 *Core*

The **core** of a fuzzy set A is the set of all points x in X such that $\mu_A(x) = 1$

$$\text{core (A)} = \{x \mid \mu_A(x) = 1\}. \tag{12.9}$$

Definition 12.4 *Normality*

A Fuzzy set A is **normal** if its core is nonempty. For a normal set, we can always find a point $x \in$ X such that $\mu_A(x) = 1$.

Definition12.5 *Crossover points*

A **crossover point** of a fuzzy set A is a point $x \in$ X at which $\mu_A(x) = 0.5$.

$$\text{crossover(A)} = \{x \mid \mu_A(x) = 0.5\} \tag{12.10}$$

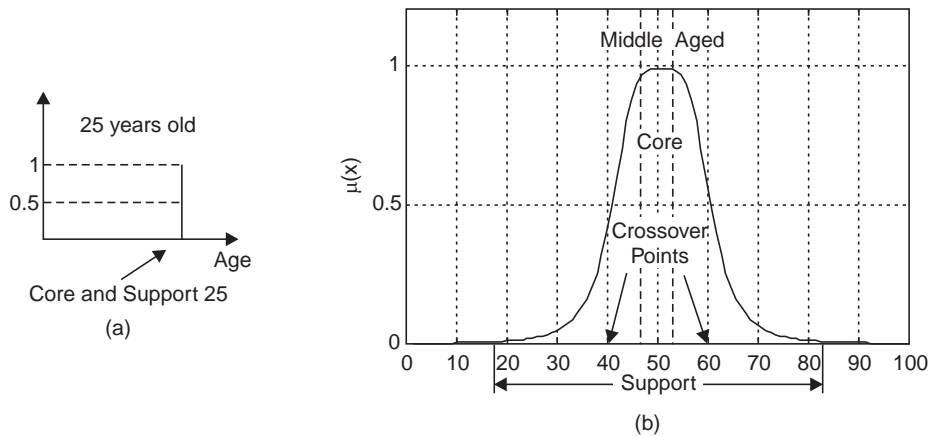



Fig. 12.4 (*a*) the fuzzy singleton "25 years old" and (*b*) core, support and crossover points of the fuzzy set "middle aged"

Definition 12.6 *Fuzzy singleton*

A fuzzy set whose support is a single point in X with $\mu_A(x) = 1$ is called a **fuzzy singleton.**

Figure 12.4(a) illustrates the fuzzy singleton characterizing "25 years old.", while Figure 12.4(b) shows the core, support, and crossover points of the bell-shaped membership function representing "middle aged".

Definition 12.7 *Convexity*

A fuzzy set A is **convex** if and only if for any $x_1, x_2 \in$ X and any $\lambda \in [0, 1]$

$$\mu_A(\lambda x_1 + (1 - \lambda)x_2) \geq \min\{\mu_A(x_1), \mu_A(x_2)\}, \quad (12.11)$$

A crisp set C in R^+ is convex if and only if for any two points $x_1 \in$ C and $x_2 \in$ C, their convex combination

$$\lambda x_1 + (1 - \lambda)x_2 \text{ is still in C, where } 0 \leq \lambda \leq 1. \quad (12.12)$$

Note that the definition of convexity of a fuzzy set is not as strict as the common definition of convexity of a function. For comparison, the definition of convexity of a function $f(x)$ is

$$f(\lambda x_1 + (1 - \lambda)x_2) \geq f(x_1) + (1 - \lambda) f(x_2)$$

which is a tighter condition than Equation (12.11). Figure 12.5 (a) shows a convex set whereas Figure 12.5(b) shows a non convex set. Convex sets are normally used to represent fuzzy numbers and are also useful for the representation of linguistic notions.

Definition 12.8 *Fuzzy numbers*

If A is a convex single-point normal fuzzy set defined on the real line R^+, then A is often termed a fuzzy number. Most non-composite fuzzy sets used in the literature satisfy the conditions of normality and convexity, so fuzzy numbers are basic types of fuzzy sets.

Definition 12.9 *Bandwidths of normal and convex fuzzy sets*

The **bandwidth** or **width** for a normal and convex fuzzy set, is defined as the distance between the two unique crossover points :

$$\text{width (A)} = |x_2 - x_1|,$$

$$\text{where } \mu_A(x_1) = \mu_A(x_2) = 0.5$$

Definition 12.10 *Symmetry*

A fuzzy set A is **symmetric** if its MF is symmetric around the point $x = c$ if

$$\mu_A(c + x) = \mu_A(c - x) \text{ for all } x \in X.$$

Definition 12.11 *Open left, open right, closed*

A fuzzy set A **is open left** if $\lim_{x \to -\infty} \mu_A(x) = 1$ and $\lim_{x \to +\infty} \mu_A(x) = 0$; **open right** if $\lim_{x \to -\infty} \mu_A(x) = 0$ and $\lim_{x \to +\infty} \mu_A(x) = 1$; and **closed** if $\lim_{x \to -\infty} \mu_A(x) = \lim_{x \to +\infty} \mu_A(x) = 0$

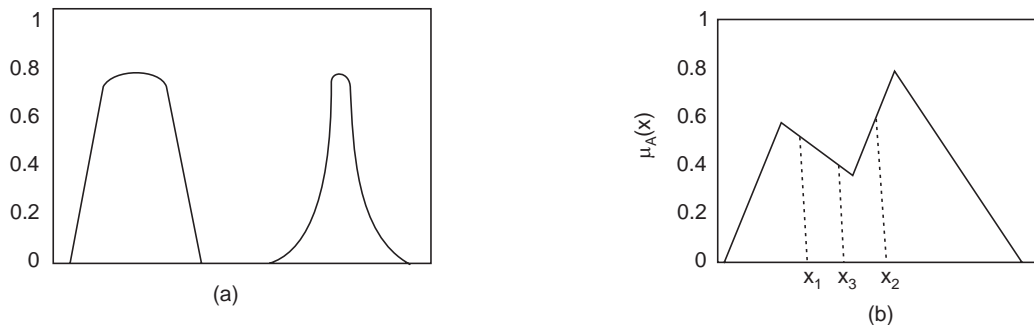



Fig. 12.5 (a) shows two convex fuzzy sets [the left fuzzy set satisfies both Equations (12.11) and (12.12), whereas the right one satisfies Equation (12.11) only]; Figure 12.5(b) is a non-convex fuzzy set.

For instance, the fuzzy set "young" in Figure 12.5 is open left; "old" is open right; and "middle aged" is closed.

12.6 SET-THEORETIC OPERATIONS

12.6.1 Classical Operators on Fuzzy Sets

Three most basic operations on classical sets are **union, intersection,** and **complement**. A number of identities that can be established by using these three operations [42-43] are listed in Table 12.1. These identities can be easily verified using Venn diagrams. Before introducing these three fuzzy set operations, we shall define the notion of containment, which plays an important role in sets—both crisp and fuzzy.

Definition 12.12 *Containment or subset*

Fuzzy set A is **contained** in fuzzy set B (or A is a **subset** of B) if and only if $\mu_A(x) \leq \mu_B(x)$ for all x. In symbols,

$$A \subseteq B \Leftrightarrow \mu_A(x) \leq \mu_B(x) \tag{12.13}$$

Figure 12.6 illustrates the concept of $A \subseteq B$

Definition 12.13 *Union (disjunction)*

The **union** of two fuzzy sets A and B is a fuzzy set C, represented as $C = A \cup B$ (*i.e.*, C = A OR B), whose MF is related to those of A and B by

$$\mu_c(x) = \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x) \tag{12.14}$$

Zadeh [2], provided a more intuitive but equivalent definition for union operation of fuzzy sets A and B as the "smallest" fuzzy set containing both A and B. Alternatively, if D is any fuzzy set that contains both A and B, then it also contains $A \cup B$.

Definition 12.14 *Intersection*

The **intersection** of two fuzzy sets A and B is a fuzzy set C, written as $C = A \cap B$ (or C = A AND B) whose MF is related to those of A and B by

$$\mu_c(x) = \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x) \tag{12.15}$$

The intersection of A and B is the "largest" fuzzy set which is contained in both A and B. This reduces to the ordinary intersection operation if both A and B are crisp.

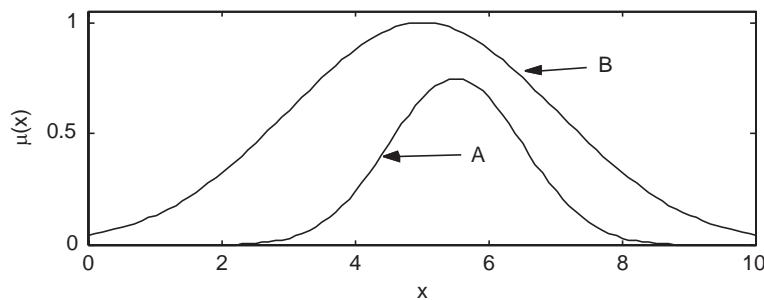



Fig. 12.6 The concept of $A \subseteq B$.

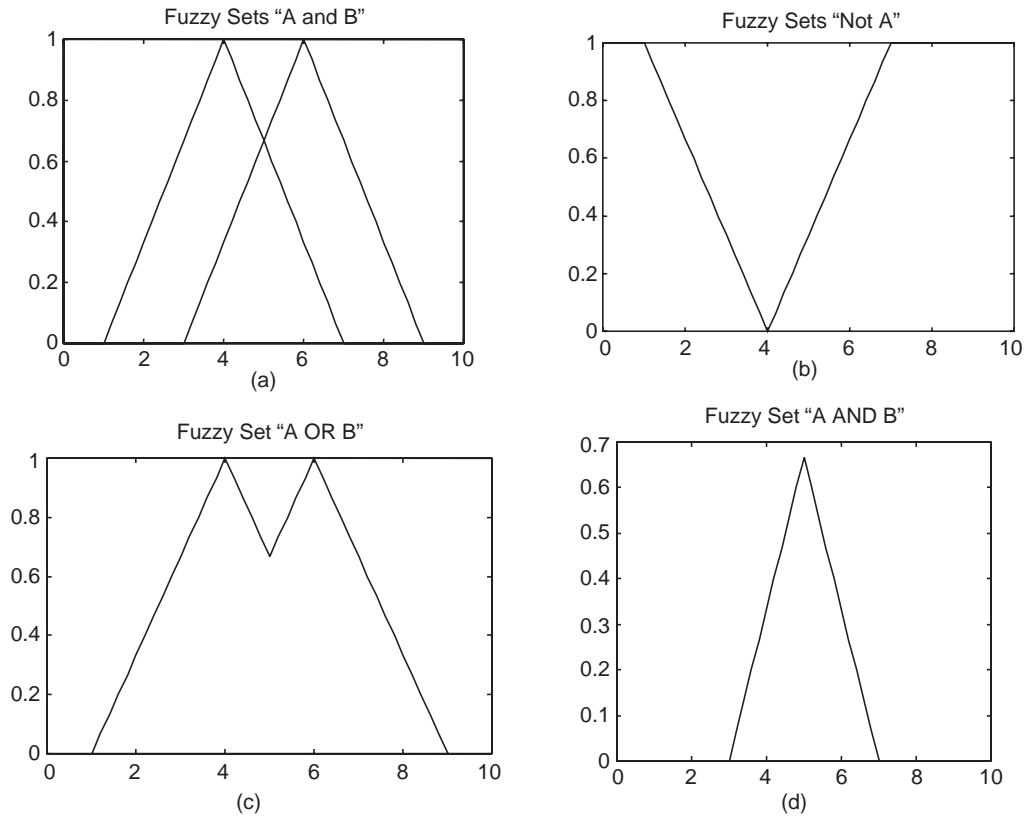



Fig. 12.7 (*a*) two fuzzy sets A and B; (*b*) complement of A; (*c*) union of A and B ; and (*d*) intersection of A and B

Definition 12.15 *Complement (negation)*

The **complement** of fuzzy set A, denoted by $\overline{A}$ (NOT A), is defined as :

$$\mu_{\overline{A}}(x) = 1 - \mu_A(x) \tag{12.16}$$

The three basic fuzzy operations: complement, union and intersection are illustrated in Fig. 12.7.

If the values of the membership functions are restricted to either 0 or 1, Equations (12.14), (12.15), and (12.16) yield the same results as the ordinary set operations. For avoiding confusion, the max [Equation (12.14)], min [Equation (12.15)], and the complement operator [Equation (12.16)] will be referred to as the **classical fuzzy operators** for intersection, union, and negation, respectively, on fuzzy sets.

Table 12.1 Basic identities of classical sets, where A, B, and C are crisp sets; $\overline{A}$, $\overline{B}$ and $\overline{C}$ are their corresponding complements; X is the universe; and ϕ is the empty set

Law of contradiction	$A \cap \overline{A} = \Phi$
Law of excluded middle	$A \cup \overline{A} = X$
Identity	$A \cap X = A$; $A \cup \Phi = A$
Idempotence	$A \cap A = A$, $A \cup A = A$
Involution	$\overline{\overline{A}} = A$

(Contd.)...

Chapter 12

Commutativity	$A \cap B = B \cap A$; $A \cup B = B \cup A$
Associativity	$(A \cup B) \cup C = A \cup (B \cup C)$ $(A \cap B) \cap C = A \cap (B \cap C)$
Distributivity	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
Absorption	$A \cup (A \cap B) = A$; $A \cap (A \cup B) = A$ $A \cup X = X$; $A \cup \Phi = A$
Absorption of complement	$A \cup (\overline{A} \cap B) = A \cup B$ $A \cap (\overline{A} \cup B) = A \cap B$
DeMorgan's laws	$\overline{A \cup B} = \overline{A} \cap \overline{B}$
	$\overline{A \cap B} = \overline{A} \cup \overline{B}$

The definition of some more set theoretic operations like cartesian product and fuzzy relation are taken up in Section 12.9.

12.6.2 Generalized Fuzzy Operators

For non fuzzy sets only one type of operation is possible for complement, union or intersection. For fuzzy sets there are other possibilities. The new fuzzy operators will be proposed on axiomatic bases. Then some relations that satisfy these axioms will be presented.

12.6.2.1 *Fuzzy Complement*

Let $C : [0, 1] \rightarrow [0, 1]$ be a mapping that transforms the membership function of fuzzy set A into the membership function of the complement of A, that is,

$$C\,[\mu_A(x)] = \mu'_{\overline{A}}(x) \tag{12.17}$$

In the case of (12.16), $C\,[\mu_A(x)] = 1 - \mu_A(x)$. In order for the function C to be qualified as a fuzzy complement, it should satisfy at least Axioms c_1 and c_2 given below:

Let a and b denote membership functions of some fuzzy sets, say, $a = \mu_A(x)$ and $b = \mu_B(x)$.

Axiom c_1 : Boundary condition : $c(0) = 1$ and $c(1) = 0$

Axiom c_2 : Monotonic Property : For all a, b $\in$ [0, 1], if $a < b$, then $c(a) \geq c(b)$; that is, c (.) is monotonic non-increasing.

Axiom c_3 : Continuity : c (.) is a continuous function.

Axiom c_4 : Involution : c (.) is involutive, which means that $c(c(a)) = a$

Axiom c_1 shows that if an element belongs to a fuzzy set to degree zero, then it should belong to the complement of this fuzzy set to degree one and vice versa. Axiom c_2 requires that an increase in membership value must result in a decrease or no change in membership value for the complement. Axiom c_4 shows that complement is involutive. Clearly, any violation of these four requirements will result in an operator that is unacceptable as a general complement.

The four Axioms are not independent, as stated (without proof) by the following theorem.

Theorem 12.1 Any function $c : [0, 1] \rightarrow [0, 1]$ that satisfies Axioms c_2 and c_4, also satisfies Axioms c_1 and c_3.

(*a*) One class of fuzzy complements is the *Sugeno class* [40] defined by

$$c_\lambda(x) = \frac{1 - x}{1 + \lambda x} \quad \text{where} \quad \lambda \in (-1, \infty). \tag{12.18}$$

For each value of the parameter λ, we obtain a particular fuzzy complement. It may be verified that the complement defined by (12.18) satisfies Axioms c_1 and c_2. Fig. 12.8 illustrates this class of fuzzy complements for different values of λ. It is to be noted that when λ = 0 it becomes the basic fuzzy complement (12.16).

12.6.2.2 *Fuzzy Union and Intersection*

Let us now present the intersection and union operations on fuzzy sets, which are often referred to as triangular norms (*t*-norms) and triangular conorms (*t*-conorms) respectively.

12.6.2.3 *Fuzzy Intersection: The T-Norm*

Let $t : [0, 1] \times [0, 1] \to [0, 1]$ be a function that transforms the membership functions of fuzzy sets A and B into the membership function of the intersection of A and B, that is,

$$t[\mu_A(x), \mu_B(x)] = \mu_{A \cap B}(x) \tag{12.19}$$

In the case of (12.15), $t[\mu_A(x), \mu_B(x)] = \min[\mu_A(x), \mu_B(x)]$. The function t will be qualified as an intersection, if it satisfies the following four requirements:

Axiom t_1 Boundary condition : $t(0,0) = 0;\ t(a, 1) = t(1, a) = a$

Axiom t_2 Commutativity : $t(a, b) = t(b, a)$

Axiom t_3 Monotonicity : If $a \le a'$ and $b \le b'$, then $t(a, b) \le t(a', b')$

Axiom t_4 Associativity : $t\,[t(a, b)\,(c)] = t\,[a, t(b, c)]$

Axiom t_1 indicates the boundary condition of the intersection function. Axiom t_2 indicates that the order in which the intersection of fuzzy sets are computed, has no effect on the result. Axiom t_3 shows a natural requirement for intersection : a decrease in membership values in the two fuzzy sets should result in an decrease in membership value in the intersection of the two fuzzy sets. Axiom t_4 allows the extensions of the intersection operations to more than two fuzzy sets.

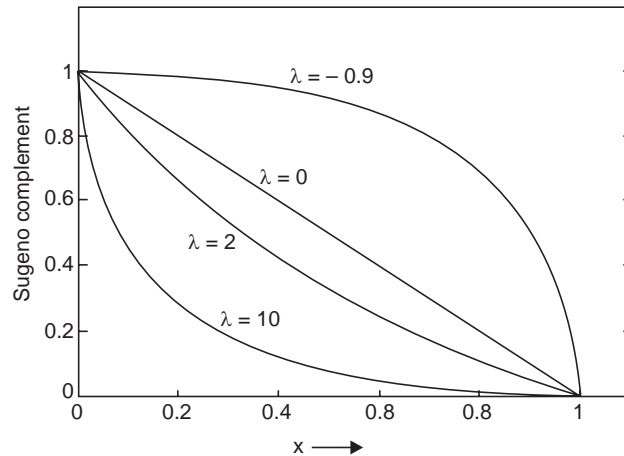



Fig. 12.8 Sugeno class of complements

Two functions given below, belong to Hamacher family that satisfy the axioms $t_1 - t_4$ and can be used as T-norms.

(*i*) Algebraic Product : $t_{ap} = ab$ (12.20*a*)

(*ii*) Hamacher *t*-norm : $t_H = \dfrac{a.b}{a + b - a.b}$ (12.20*b*)

(*iii*) Einstein product : $t_{ep} = \dfrac{ab}{2-(a+b-ab)}$ (12.20*c*)

(*iv*) Yager [41] class : $t_y(a, b) = 1 - \min\,[1, \{(1-a)^c + (1-b)^c\}^{1/c}]$, where $c \in (0, \infty)$ (12.20*d*)

t_H is used in control system to combine several fuzzy sets that are not so optimistic as the minimum operation.

12.6.2.4 *Fuzzy Union: The T-Conorm (or S-Norm)*

Let s : [0 , 1] × [0, 1] → [0, 1] be a mapping function that transforms the membership functions of fuzzy sets A and B into the membership function of the union of A and B, that is,

$$s[\mu_A(x), \mu_B(x)] = \mu_{A \cup B}(x) \quad (12.21)$$

The *t*-conorm function will qualify as an union, if it satisfies the following four requirements:

Axiom s_1 Boundary condition : $s(1, 1) = 1, s(0, a) = s(a, 0) = a$

Axiom s_2 Commutative condition : $s(a, b) = s(b, a)$

Axiom s_3 Monotonicity condition : If $a \le a'$ and $b \le b'$, then $s(a, b) \le s(a', b')$

Axiom s_4 Associative condition : $s(s(a, b), c) = s(a, s(b, c))$

Axiom s_1 indicates the boundary condition of the union function. Axiom s_2 indicates that the order in which the fuzzy sets are combined has no effect on the result. Axiom s_3 shows a natural requirement for union: an increase in membership values in the two fuzzy sets should result in an increase in membership value in the union of the two fuzzy sets. Axiom s_4 allows the extensions of the union operations to more than two fuzzy sets.

For any *t*-norm, there is an *t*-conorm associated with it and vice versa.

Definition 12.16. Any function $s : [0, 1] \times [0, 1] \to [0, 1]$ that satisfies Axioms $s_1 - s_4$ is called an *t*-conorm.

It may be easily proved that the basic fuzzy union max of (12.14) is a *t*-conorm. Two classes of *t*-conorms of the Hamachar class are listed below:

(*i*) Algebraic sum : $s_{as}(a, b) = a + b - ab$ (12.22*a*)

(*ii*) Einstein sum : $s_{es}(a, b) = \dfrac{a+b}{1+ab}$ (12.22*b*)

(*iii*) Yager class : $s_y(a, b) = \min\,[1, (a^c + b^c)^{1/c}]$, where $c \in (0, \infty)$ (12.22*c*)

12.7 MF FORMULATION AND PARAMETERIZATION

A fuzzy set is completely characterized by its MF, but how to determine the membership functions ? Since most fuzzy sets in use have a universe of discourse X consisting of the real line R, it would be impractical to list all the pairs defining a membership function. Two approaches have been found to be useful for making a choice of a membership function out of a myriad of alternatives.

The first approach is to use the knowledge of human experts to specify the membership functions in a particular domain. Since fuzzy sets are often used to capture human knowledge, the membership functions represent a part of human knowledge. By this approach one can get a rough idea of the membership function, which needs fine-tuning. In the second approach, the membership functions may be determined from the data collected from various sensors. A structure of the membership functions in the form of a mathematical formula (as in Example 12.3) may be created and then the data may be used to fine-tune the parameters of the membership functions. Both the approaches are used in practice.

In this section a classes of parameterized functions commonly used to define MFs for one dimension will be described. Membership functions of higher dimensions can similarly be defined.

12.7.1 MFs of One Dimension

Several classes of parameterized MF's with single input are defined first.

Definition 12.17 Triangular MF's

$$\text{triangle}(x, a, b, c) = \begin{cases} 0 & x \le a \\ \dfrac{x-a}{b-a} & a \le x \le b \\ \dfrac{c-x}{c-b} & b \le x < c \\ 0 & c \le x \end{cases} \quad (12.23)$$

An alternative expression for the preceding equation may be obtained by using min and max operation:

$$\text{triangle}(x\,;a, b, c) = \max\left(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0\right) \quad (12.24)$$

where the three corners of the triangular function are determined by the parameters $\{a, b, c\}$ satisfying the constraint $a < b <$ c. Figure 12.9(a) illustrates a triangular MF defined by triangle (x; 2, 5, 8)

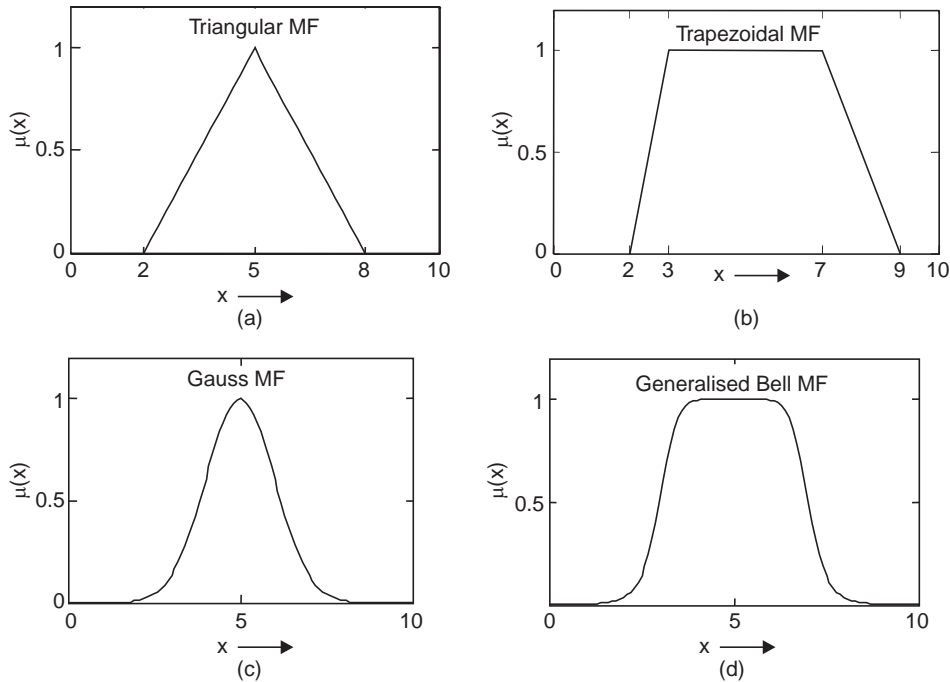



Fig. 12.9 Examples of four classes of parameterized MFs: (a) triangle (x, 2, 5, 8); (b) trapezoid (x, 2, 3, 7, 9); (c) Gaussian (x, 1, 5); (d) bell (x, 2, 4, 5).

Definition 12.18 Trapezoidal MFs

A **trapezoidal** MF, specified by min and max operators on four parameters $\{a, b, c, d\}$ is given by :

$$\text{trapezoid}\,(x;\, a, b, c, d) = \max\left(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0\right) \tag{12.25}$$

The alternative expression using four parameters $\{a, b, c, d\}$ for the trapezoidal MF is given by :

$$\text{trapezoid}\,(x\,;\, a, b, c, d) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{b-a}, & a \le x \le b \\ 1, & b \le x \le c \\ \dfrac{d-x}{d-c}, & c \le x \le d \\ 0 & d \le x \end{cases} \tag{12.26}$$

The x coordinates of the four corners of the trapezoidal membership functions are determined by the four parameters $\{a, b, c, d\}$ satisfying the constraint with $a < b \le c < d$. It is to be noted that a trapezoid MF reduces to a triangular MF when the parameter b is equal to c in relations (12.25) and (12.26). Figure 12.9(b) illustrates a trapezoid MF defined by trapezoid (x, 2, 3, 7, 9).

Both the triangular MFs and trapezoidal MFs have been extensively used in real-time implementations of fuzzy controllers because of the computational efficiency arising out of their simple functional forms. However, none of these membership functions are smooth at the corner points specified by the parameters. Some MFs, which are smooth, are introduced below:

Definition 12.19 Gaussian Membership Functions

A Gaussian MF is completely specified by two parameters $\{c, \sigma\}$:

$$\text{gaussian}\,(x;\, c, \sigma) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{12.27}$$

where c represents the MF's center and σ determines the MF's width. Figure 12.9 (c) plots a Gaussian MF defined by Gaussian (x; 1, 5).

Definition 12.20 Generalized bell Membership Functions

A generalized bell MF (or bell MF) is specified by three parameters $\{a, b, c\}$:

$$\text{bell}\,(x, a, b, c) = \frac{1}{1 + \left|\dfrac{x-c}{a}\right|^{2b}} \tag{12.28}$$

Here the parameter b is taken to be positive since negative value will make the shape of this MF an upside-down bell. It is also referred to as the Cauchy MF, since this MF is a direct generalization of the Cauchy distribution used in probability theory. The parameters c and a are adjusted to vary the center and width of the MF, and b is used to control the slopes at the crossover points. Figure 12.9 (d) illustrates a generalized bell MF defined by bell (x; 2, 4, 5).

Even though Gaussian and bell membership functions are smooth, they cannot represent asymmetric characteristics required in some control situations. The sigmoidal functions, extensively used in neural networks as activation function, can serve this purpose.

Definition 12.21 Sigmoidal Membership Functions

A sigmoidal membership function is expressed in terms of two parameters (a, c) as

$$\text{sig}\ (x, a, c) = \frac{1}{1 + e^{[-a(x-c)]}} \tag{12.29}$$

The parameter a is used to control the slope of sigmoidal function at the cross-over point $x = c$.

When the parameter **a** is positive the sigmoidal function becomes open right and a negative value for **a** makes it open to the left corresponding to “big positive” and “big negative” values of x respectively. Figure 12.10(a, b) shows sig (x, – 2, 5) and sig (x, 2, 5) respectively, while the membership function in Figure 12.10(c) is the sum of sig (x, 2, 5) and sig (x,– 2, 10), which is symmetric.

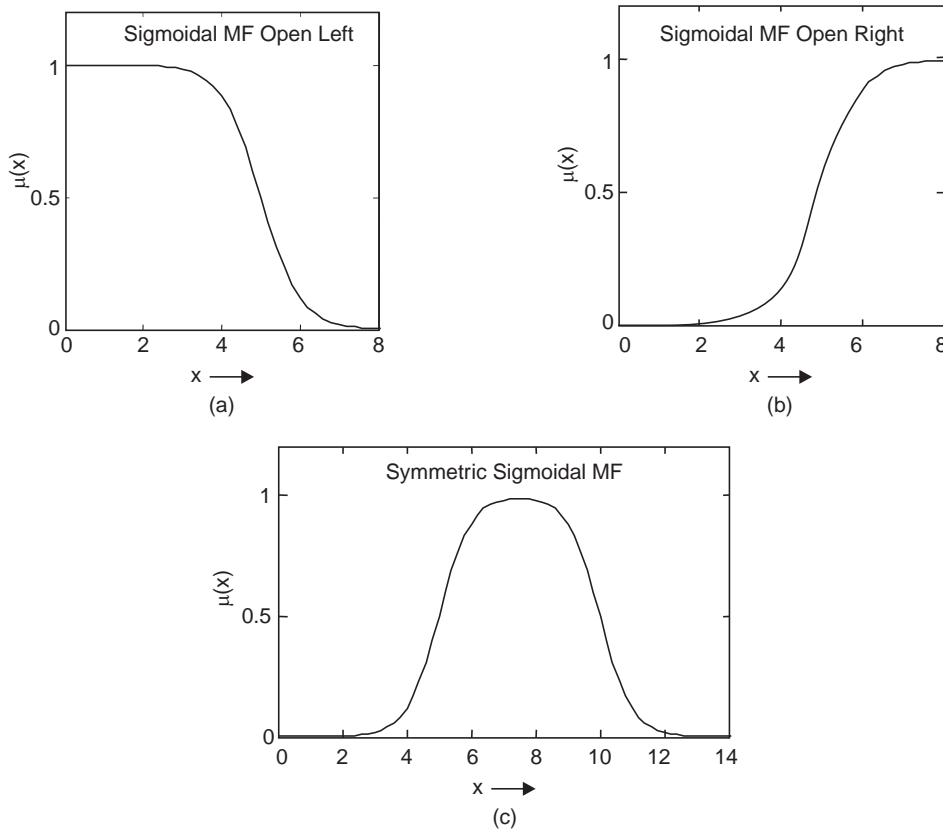



Fig. 12.10 Sigmoidal membership functions ; (a) sig(x, – 2, 5) (b) sig(x, 2, 5) (c) sig(x, 2, 5) + sig(x, – 2, 10)

Similarly, by extending the concepts presented in this section, we can have n-dimensional MFs.

12.8 FROM NUMERICAL VARIABLES TO LINGUISTIC VARIABLES

In our everyday life, words are routinely used to describe variables. When we say “water is hot,” we mean “water’s temperature is high”. Here the word “high” is used to describe the variable “water’s temperature.” That is, the variable “water’s temperature” takes the term

"high" as its value. Clearly, the variable "water's temperature" also can take numbers like 60°C, 85°C, etc., as its values. When a variable takes numbers as its values, the mathematical framework is well equipped to manipulate it. But when a variable takes words as its values, the classical mathematical theory is ill equipped for its manipulation. In order to provide such a formal framework, the concept of linguistic variables was introduced. Roughly speaking, when a variable takes words from natural languages as its values, it is referred to a linguistic variable. Fuzzy sets are used to characterise the words so as to put them in a mathematical framework. So we offer the following definition.

Definition 12.22 If a variable can take words from natural languages as its values, it is called a linguistic variable, where the words are characterised by fuzzy sets defined in the universe of discourse in which the variable is defined.

Example 12.6 The speed of a car is a variable x that takes values in the interval [0, 100], where 100 is the maximum speed of the car. We now define three fuzzy sets "slow," "medium," and "fast" in [0, 100] as shown in Fig. 12.11. If we view x as a linguistic variable, then it can take "slow," "medium" and "fast" as its values. That is, we can say "x is slow," "x is medium," and "x is fast" . Of course, x also can take numbers in the interval [0, 100] as its values, for example, x = 20 Kmph, 50 Kmph, 85 Kmph etc.

Definition 12.22 gives a simple and intuitive definition for linguistic variables. A more formal definition of linguistic variables as is used in fuzzy theory literature [74] is given below:

Definition 12.23 A linguistic variable is characterized by *(X, T, U, M),* where:

- X is the name of the linguistic variable; in Example 12.6, X is the speed of the car.
- *T* is the set of linguistic values, called term set, that X can take ; in Example 12.6, T = {slow, medium, fast}.
- U is the actual physical domain in which the linguistic variable X takes its quantitative (crisp) values ; in Example 12.6, U = [0, 100].
- M is a semantic rule that relates each linguistic value in T with a fuzzy set in U ; in Example 12.6, M relates "slow," "medium," and "fast" with the membership functions shown in Fig. 12.11.

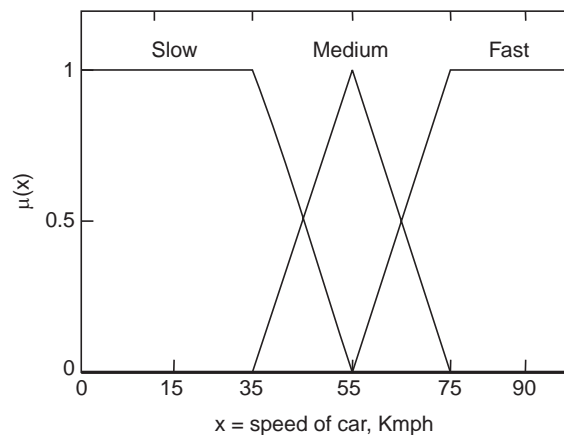



Fig. 12.11 The speed of a car as a linguistic variable represented by fuzzy sets "slow", "medium" and "fast".

Comparing definitions 12.22 with 12.23, we see that they are essentially equivalent. Definition 12.22 is more intuitive while definition 12.23 looks more formal. From these

definitions it is clear that linguistic variables are extensions of numerical variables in the sense that they are allowed to take fuzzy sets as their values. The concept of linguistic variable is important because linguistic variables are the most fundamental elements in human knowledge representation. The response of sensors are expressed in numbers while the response from a human experts are expressed in words. For instance if a radar gun is used to measure the speed of a car, it gives us numbers like 50 Kmph, 52 Kmph, etc., and in response to a query about the speed of a car, humans give words like "it's slow," "it's fast," etc. Hence, by introducing the concept of linguistic variables, one can formulate vague descriptions in natural languages in precise mathematical terms. This is the first step to incorporate human knowledge into engineering systems in a systematic and efficient manner.

12.8.1 Term Sets of Linguistic Variables

Consider the schedule of the membership functions of the fuzzy set for a linguistic variable x in the form shown in Figure 12.12, in which the fuzzy sets are labelled with linguistic variables Negative-Big (NB), Negative-Medium (NM), Negative-Small (NS), Zero (ZE), Positive-Small (PS), Positive-Medium (PM), and Positive-Big (PB). The terms NB, ZE, PB etc., are the names of the linguistic variables, which are used to partition the input domain x normalised to [– 10 + 10] and the set x = {NB, NM, NS, ZE, PS, PM, PB} $x \in$ U is known as term set for the input x. In general for a typical fuzzy logic control system, the input $\mathbf{x}$ may be a n-dimensional vector and output y may be m dimensional vector, in which case the linguistic variables x_i, form a input space $U = U_1 \times U_2 \times \ldots U_n$ and the output linguistic variables y_i form a output space $V = V_1 \times V_2 \times \ldots V_m$

So the linguistic variables x_i in the universe of discourse U_i may be characterized by the term set T(x) with membership $\mu(x)$ where

$$T(x_i) = \left\{T_{x_i}^1, T_{x_i}^2, \ldots T_{x_i}^{k_i}\right\} \text{ and } \mu(x_i) = \left\{\mu_{x_i}^1, \mu_{x_i}^2, \ldots \mu_{x_i}^{k_i}\right\}, \text{ for } i = 1, 2, 3, \ldots n \qquad (12.30)$$

Here $T(x_i)$ is the term set of x_i, (the set of names of linguistic values of x_i) and with each value of $T_{x_i}^{k_i}$ is a fuzzy number with membership function $\mu_{x_i}^{k_i}$ defined on U_i. So $\mu(x_i)$ is a semantic rule for associating each value with its meaning. For example, if x_i indicates temperature then $T(x_i) = \left\{T_{x_i}^1, T_{x_i}^2, T_{x_i}^3\right\}$ may be "low", "medium" and "high" temperature. Similarly, an output linguistic variable y_i is associated with the term set $T(y_i) = \left\{T_{y_i}^1, T_{y_i}^2, \ldots T_{y_i}^{l_i}\right\}$ and $\mu(y_i) = \left\{\mu_{y_i}^1, \mu_{y_i}^2, \ldots \mu_{y_i}^{l_i}\right\}$ for $i = 1, 2, 3, \ldots m$

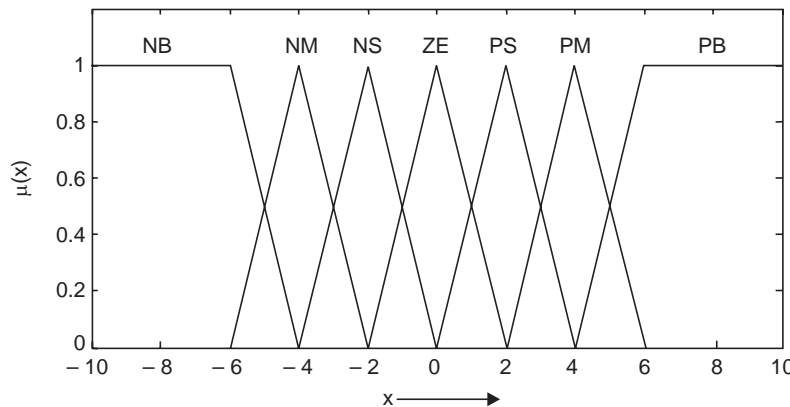



Fig. 12.12 Membership functions of the fuzzy sets NB, NM, NS, ZE, PS, PM and PB

Therefore, the input vector **x**, constituted by the linguistic variables x_i and output state vector **y** constituted by the linguistic variables y_i for a fuzzy logic controller can be defined, respectively, as

$$x = \{(x_i, \mu_i(x_i)) \mid x_i \in U_i, i = 1, 2 \ldots n\} \quad (12.31)$$
$$y = \{(y_i, \mu_i(y_i)) \mid y_i \in V_i, i = 1, 2 \ldots m\}$$

The size (or cardinality) of a term set $| T(x_i) | = k_i$ is equal to the number of fuzzy partition of x_i. The fuzzy partition determines, the granularity of the control obtainable from a Fuzzy Logic Controller (FLC). The fuzzy input space in Fig. 12.12 is divided into 7 overlapping grids in the normalized universe [– 10, + 10], so its cardinality is 7. The same normalized universe [– 10, + 10] could have been divided in 5 or 3 partitions giving a cardinality of 5 and 3 respectively. It will be shown in the next chapter that the fuzzy partitions in a fuzzy input space determine the maximum number of fuzzy control rules in fuzzy logic controller. For example, in the case of a two-input one-output fuzzy logic control system, if $| T(x_1) | = 3$ and $| T(x_2) | = 7$, then the maximum number of fuzzy control rules is $3 \times 7 = 21$. The input membership functions $\mu_{x_i}^{k}$, $\mathbf{k} = 1, 2, 3 \ldots, k_i$ and the output membership functions $\mu_{y_i}^{1}$, $l = 1, 2, 3 \ldots, l_i$ used in a FLC are usually parametric functions such as triangular, trapezoidal, and bell-shaped, symmetric sigmoidal membership functions (vide Section 12.7).

12.9 CLASSICAL RELATIONS AND FUZZY RELATIONS

Relations are extensively used in logic, approximate reasoning, pattern recognition and control engineering among other disciplines of science and engineering. Before we introduce the concept of relation, let us consider some terminologies.

12.9.1 Cartesian Product

An ordered set of n elements written in the form ($a_1\, a_2\, a_3\, a_4 \ldots a_n$) is called an ordered n-tuple whereas an unordered n-tuple has no restriction on their place in the tuple and simply called a collection of n elements. For the case with $n = 2$, the n-tuple is referred to as ordered pair. For crisp sets $A_1, A_2, A_3, \ldots A_n$, the set of n-tuples $(a_1\, a_2\, a_3\, a_4 .. a_n)$, where $a_1 \in A_1$, $a_2 \in A_2$, $a_3 \in A_3$, ... and $a_n \in A_n$, is called the cartesian product of $A_1, A_2, A_3 \ldots A_n$ and is denoted by $A_1 \times A_2 \times A_3 \ldots \times A_n$. A subset of the Cartesian product of $A_1 \times A_2 \times A_3 \ldots \times A_n$ is called an *n-ary relation* over $A_1, A_2, A_3 \ldots A_n$. The cartesian product of two or more sets is not the same thing as the arithmetic product of the elements of two or more sets. For example, in the binary case with $n = 2$, Cartesian product involves pairing of elements among the two sets, whereas the arithmetic product involves the multiplication of the elements of the two sets. In the case when all the sets A_n are identical and equal to A, the cartesian product is written as A^n.

Example 12.7 Let $A = \{a_1, a_2\}$ and $B = \{ b_1, b_2, b_3\}$ be two sets. The various cartesian products of these two sets may be written as :

$A \times B = \{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_2, b_1), (a_2, b_2), (a_2, b_3)\}$

$B \times A = \{(b_1, a_1), (b_1, a_2), (b_2, a_1), (b_2, a_2), (b_3, a_1), (b_3, a_2)\}$

$A \times A = A^2 = \{(a_1, a_1), (a_1, a_2), (a_2, a_1), (a_2, a_2)$ and

$B \times B = B^2 = \{(b_1, b_1), (b_1, b_2), (b_1, b_3), (b_2, b_1), (b_2, b_2), (b_2, b_3), (b_3, b_1), (b_3, b_2), (b_3, b_3)\}$

12.9.2 Crisp Relations

A subset of the cartesian product $A_1 \times A_2 \times A_3 \ldots \times A_n$ is called an n-array relation over $A_1, A_2, A_3, A_4 \ldots A_n$. The cartesian product of two universes X and Y, written as

$$X \times Y = [(x, y) \mid x \in X, y \in Y]$$

represents an unconstrained relation between every element of $x \in$ X with every element of $y \in$ Y, giving rise to ordered pair (x, y). In a constrained relation, not all elements in X are related with every elements of Y. The strength of this relationship between ordered pair of elements is denoted by the membership function given below:

$$\mu_{x \times y}(x, y) = \begin{cases} 1, & (x, y) \in X \times Y \\ 0, & (x, y) \notin X \times Y \end{cases} \tag{12.32}$$

When the universe or sets are finite the relation can be conveniently represented by an n-dimensional matrix, called a relation matrix.

Example 12.8 Let us consider a 3-dimensional binary relation R of languages spoken (X), currency used (Y) in a country (Z), where X = {English, French}, Y = {Dollar, Franc, Rupees} and Z = [USA, France, Canada, India}.

R(X, Y, Z) = {(English, Dollar, USA), (English, Dollar, Canada),(English, Rupees, India), (French, Franc, France), (French, Franc, Canada).].

This relation can also be written in a 3-dimensional membership array as :

	USA	France	Canada	India
Dollar	1	0	1	0
Franc	0	0	0	0
Rupees	0	0	0	1

English

	USA	France	Canada	India
Dollar	0	0	1	0
Franc	0	1	0	0
Rupees	0	0	0	0

French

For convenience of computer processing, a 5-dimensional array L × L(L $\in N_5$) can be represented as a library x_1(5-dimensional) with x_{2i} books (i = 1 to 4), each book with x_{3j} pages (j = 1 to 3) and each page with x_{4k} matrices (k = 1 to 2), each matrix with x_m rows and each row with n elements.

Relations can also be defined in continuous domain.

Example 12.9 Consider a 2-dimensional relation expressed as :

$$R = \{(x, y) \mid y \geq 5x, x \in X, y \in Y\}$$

This relation can also be expressed as membership function

$$\mu_R(x, y) = \begin{cases} 1 & y \geq 5x \\ 0 & y < 5x \end{cases}$$

12.9.3 Fuzzy Relations

We shall get a fuzzy relation by allowing the membership grades of the crisp relations in discrete tuples to assume values in [0, 1]. Thus a fuzzy relation is a fuzzy set defined on the cartesian product of crisp sets A_1, A_2, A_3, ... A_n when tuples (a_1 a_2 a_3 a_4 ... a_n) have varying degrees of membership grades within the relation. The membership grade indicates the strength of the relation among the elements of the tuple. A fuzzy relations can also be represented by an n-dimensional membership array whose elements correspond to n-tuples in the universal set.

These entries take on values representing the membership grades of the corresponding n-tuples.

Definition 12.24 A fuzzy relation is a fuzzy set defined in the Cartesian product of crisp sets $U_1, U_2, \ldots U_n$.

The fuzzy relation Q in $U_1 \times U_2 \times \ldots \times U_n$, is defined as the fuzzy set

$$Q = \{((u_1, u_2, \ldots, u_n), \mu_Q (u_1, u_2, \ldots, u_n)) / (u_1, u_2, \ldots, u_n) \in U_1 \times U_2 \times \ldots U_n\} \quad (12.33)$$

A binary fuzzy relation is a special fuzzy set defined in the Cartesian product of two crisp sets. A binary relation on a finite Cartesian product is usually represented by a fuzzy relational matrix, whose elements are the membership values of the corresponding pairs belonging to the fuzzy relation.

Example 12.10. Let U and V be the set of real numbers, that is, U = V = R. A fuzzy relation "x is approximately equal to y," denoted by AE, may be defined by the membership function

$$\mu_{AE}(x, y) = e^{-(x-y)^2} \quad (12.34)$$

Similarly, a fuzzy relation "x is much larger than y", denoted by ML, may be defined by the membership function

$$\mu_{ML}(x, y) = \frac{1}{1 + e^{-(x-y)}} \quad (12.35)$$

Of course, one may choose other membership functions to represent these fuzzy relations.

Example 12.11 Let U and V be two sets of cities where U = {Paris, Calcutta, Dacca} and V = {London, Calcutta}. Let us define the relational concept "very far" between these two sets of cities. Clearly, classical relations are not useful here because the concept "very far" is not well-defined in the framework of classical sets and relations. If we use a number in the interval [0, 1] to represent the degree of "very far" then the concept "very far" may be represented by the following list notation

R(X × Y) = 0.25/(London, Paris) + 0.95/(London, Calcutta) + 1.0/(London, Dacca)
+ 0.9/ (Calcutta, Paris) + 0.0/(Calcutta, Calcutta) + 0.2/(Calcutta, Dacca)

Or by the fuzzy relational matrix:

		V	
		London	Calcutta
	Paris	0.25	0.9
U	Calcutta	0.95	0
	Dacca	1.0	0.2

(12.36)

In Example 12.11 the fuzzy relation has been used to extend the concept of crisp relation and may be used to formulate more relationships in real world problems.

12.9.4 Operation on Fuzzy Relations

Fuzzy relations are very important in fuzzy control because they can describe interaction between state variables. Since fuzzy relation is also a fuzzy set, all the operations applicable to a set, like union, intersection, complement and containment can also be performed on the fuzzy relations. This is defined below for the case n = 2 with relation R and S on the cartesian space X × Y and can be easily extended to n-array tuples.

Definition 12.25

(*a*) Union $\mu_{R \cup S}(x, y) = \max [\mu_R(x, y), \mu_S(x, y)]$

(*b*) Intersection $\mu_{R \cap S}(x, y) = \min [\mu_R(x, y), \mu_S(x, y)]$

(*c*) Complement $\mu_{\overline{R}}(x, y) = 1 - \mu_R(x, y)$

(*d*) Containment $R \subset S \Rightarrow \mu_R(x, y) \le \mu_S(x, y)$

In the above definitions any *s*-norm for union, any *t*-norm for intersection and any other complement could have been used instead of ones used there.

Example 12.12. Let us consider two relations matrices R and S representing fuzzy membership grades of two sets of cities U and V in the cartesian product space X × Y.

R = *x* is very far from *y*

U \ V	London	Calcutta
Paris	0.25	0.9
Calcutta	0.95	0
Dacca	1.0	0.2

(12.37)

S = *y* is near to *x*

U \ V	London	Calcutta
Paris	0.80	0.40
Calcutta	0.40	1.00
Dacca	0.35	0.95

(12.38)

So the representation of the perception of "*x* is far from *y*" and "*y* is near to *x*" can be obtained by taking the intersection of the relation R and S

$R \cap S =$

U \ V	London	Calcutta
Paris	0.25	0.40
Calcutta	0.40	0
Dacca	0.35	0.2

We could have obtained a different result using the Hamacher t-norm of relation (12.20b) as shown below:

$R \cap S =$

U \ V	London	Calcutta
Paris	0.2353	0.3830
Calcutta	0.3918	0
Dacca	0.35	0.1979

The representation of the perception of "x is far from y" or "y is near to x" can be obtained by taking the union of the relation R and S and is given below:

$R \cup S =$ U

	V: London	Calcutta
Paris	0.80	0.90
Calcutta	0.95	1.00
Dacca	1.00	0.95

If we use algebraic sum t-conorm for computing the union operation $s_{as} = a + b - ab$ (see relation 12.22a), the result is given by:

$R \cup S =$ U

	V: London	Calcutta
Paris	0.85	0.94
Calcutta	0.97	1.00
Dacca	1.00	0.96

It is found that the elements of the union $R \cup S$ with algebraic t-conorm, s_{as} are at least as high as with max operation but are more optimistic than obtained by max operation. The operator "or" in the above union is interpreted as "inclusive or" that is "a or b or a and b" and does not imply "either a or b exclusively".

Projection and cylindrical extensions are two important operations on fuzzy sets and fuzzy relations. The projection operation decreases the dimension of a tuple—it reduces a ternary relation to a binary relation, a binary relation to a fuzzy set and a fuzzy set to a single crisp value. Since the concept of projection is complex, we first illustrate it with examples for the $n = 2$ before taking up the formal definition.

Let us rewrite the relation in (12.36) in general form

R = x is very far from y (12.39)

	y_1	y_2
x_1	0.25	0.9
x_2	0.95	0
x_3	1.0	0.2

Projection on X means

x_1 is assigned maximum value of first row, $x_1 = 0.9$

x_2 is assigned maximum value of second row, $x_2 = 0.95$

x_3 is assigned maximum value of third row, $x_3 = 1.0$

so that projection of R on X becomes a fuzzy set A

$$\text{proj R on X} = R \downarrow X = A = \frac{0.9}{x_1} + \frac{0.95}{x_2} + \frac{1}{x_3} \tag{12.40}$$

Similarly we can have projection on Y. It is given by the fuzzy set B in this case

$$\text{proj R on Y} = R \downarrow Y = B = \frac{1}{y_1} + \frac{0.9}{y_2} \tag{12.41}$$

We can also take projection of the set A or B, in both the cases it becomes a discrete value equal to 1.

The cylindrical extension is opposite to projection. It increases the dimension of the tuple.

Example 12.13. Consider the fuzzy set in relation (12.40).

The cylindrical extension of this fuzzy set on y gives the relation

ce of A on $y = A \uparrow y =$

	y_1	y_2
x_1	0.9	0.9
x_2	0.95	0.95
x_3	1	1

(12.42)

Similarly $B \uparrow x =$

	y_1	y_2
x_1	1.0	0.9
x_2	1.0	0.9
x_3	1.0	0.9

(12.43)

Example 12.14. Let us consider a ternary ($n = 3$) relation $R(X_1, X_2, X_3)$ in the Cartesian product space $X_1 \times X_2 \times X_3$ with $X_1 = \{x_{11}, x_{12}\}$, $X_2 = \{x_{21}, x_{22}\}$, and $X_3 = \{x_{31}, x_{32}\}$ where R is given by:

$$R(X_1, X_2, X_3) = \frac{0.4}{x_{11}x_{21}x_{32}} + \frac{0.6}{x_{11}x_{22}x_{31}} + \frac{0.2}{x_{12}x_{21}x_{32}} + \frac{0.8}{x_{12}x_{22}x_{31}} + \frac{1}{x_{12}x_{22}x_{32}}$$

From three dimensional relation R, we can derive the following two dimensional projections:

$$R_{12} = R \downarrow (X_1, X_2) = \max_{x_{1i} \in X_1,\, x_{2j} \in X_2} \left(\frac{0.4}{x_{11}x_{21}} + \frac{0.6}{x_{11}x_{22}} + \frac{0.2}{x_{12}x_{21}} + \frac{0.8}{x_{12}x_{22}} + \frac{1}{x_{12}x_{22}} \right)$$

Using max (0.8, 1) = 1 for the membership of $(x_{12}x_{22})$, we get

$$R_{12} = \frac{0.4}{x_{11}x_{21}} + \frac{0.6}{x_{11}x_{22}} + \frac{0.2}{x_{12}x_{21}} + \frac{1}{x_{12}x_{22}}$$

Similarly, $$R_{13} = R \downarrow (X_1, X_3) = \max_{x_{1i} \in X_1,\, x_{3k} \in X_3} \left(\frac{0.4}{x_{11}x_{32}} + \frac{0.6}{x_{11}x_{31}} + \frac{0.2}{x_{12}x_{32}} + \frac{0.8}{x_{12}x_{31}} + \frac{1}{x_{12}x_{32}} \right)$$

$$= \frac{0.4}{x_{11}x_{32}} + \frac{0.6}{x_{11}x_{31}} + \frac{0.8}{x_{12}x_{31}} + \frac{1}{x_{12}x_{32}}$$

and $$R_{23} = R \downarrow (X_2, X_3) = \max_{x_{2j} \in X_2,\, x_{3k} \in X_3} \left(\frac{0.4}{x_{21}x_{32}} + \frac{0.6}{x_{22}x_{31}} + \frac{0.2}{x_{21}x_{32}} + \frac{0.8}{x_{22}x_{31}} + \frac{1}{x_{22}x_{32}} \right)$$

$$= \frac{0.4}{x_{21}x_{32}} + \frac{0.8}{x_{22}x_{31}} + \frac{1}{x_{22}x_{32}}$$

The utility of the cylindrical extension can be understood from the following situation. Let A be a set in the universe X and R be a relation on X × Y and we are interested to take the intersection of the set A with the matrix R. As such, we can not take the intersection of a vector A with a matrix R. But if we extend the vector A to X × Y to form a matrix, we can take their intersection. Given a fuzzy relation S in U × V, we can define a relation $R(u, v, w) = S(u, v)$ on U × V × W, so as to get a new fuzzy set S × W which is known as the **cylindrical extensions** of S.

Cylindrical extension: The Cartesian product can be used to define cylindrical extension. Let a fuzzy set be represented as a vector of membership values where each element of the set is associated with a membership function. In vector format, the fuzzy set A in the universe X can be written as:

$$A = [\mu(x_1), \mu(x_2), \ldots \mu(x_n)] = \{\mu(x_i)\}, i = 1, 2, \ldots n$$

The cylindrical extension of A on the axis Y = $\{y_1, y_2, y_3, \dots y_m\}$ is a subset (relation) of the Cartesian product of the set A and Y in the whole Cartesian product space X × Y and is obtained as

$$\mu_R(x, y) = [A \uparrow Y] = A \times Y, \text{ where } \mu(y_j) = 1, \text{ for } j = 1, 2, 3 \dots m$$

The elements of R(x, y) are found as $\mu_R(x_i, y_j) = \mu(x_i)\, t\mu(y_j)$ for $\forall x_i \in X$ and $\forall x_i \in Y$, where t is any t-norm, including the min operator. Using the min operator, the cylindrical extension of A on Y becomes

$$\mu_R(x_i, y_j) = \mu(x_i), \text{ for } \forall x_i \in X \text{ and } \forall y_j \in Y.$$

We now present the formal definition of **projection.**

Definition 12.26 Projection

In general, let R be a fuzzy relation in $U_1 \times U_2 \times U_3 \dots \times U_n$ and let I = $\{i_1, i_2, i_3, \dots i_k\}$ be a subset of {1, 2, 3, ... n} with $\{i_1 < i_2 < i_3 < \dots < i_k\}$. The projection of R onto $U_{i1} \times U_{i2} \times U_{i3} \dots \times U_{ik}$ is defined to be fuzzy subset of $U_1 \times U_2 \times U_3 \dots \times U_n$ given by

$$S(u_{i1}, u_{i2}, u_{i3} \dots u_{ik}) = \bigcup_{u_j,\, j \notin I} \{R(u_1, u_2, u_3 \dots u_n)\}$$

For example if R is a fuzzy relation on U × V × W, the projection on U × V is given by $S(u, v) = U_{w \in W} R(u, v, w)$

The cylindrical extension is, in some sense, an inverse to the projection. The combination of fuzzy sets and fuzzy relation with the help of cylindrical extensions and projection is called the fuzzy composition.

Definition 12.27 Let A be a fuzzy set in X and R be a fuzzy relation in X × Y The composition of the set A with the fuzzy relation R (written as A o R) will produce a set B in Y given by

$$B = A \text{ o } R = \text{proj of } (ce\ (A) \cap R) \text{ on } Y$$

Now if *min* is used for implementing *intersection* and *max* for *projection* the membership function for B may be obtained as

$$\mu_B(y) = \max_x \min[\mu_A(x), \mu_R(x, y)] \quad (12.44)$$

This is called *max min composition.* If *product* is used for *intersection* and *max* for *projection* then the membership function for B is obtained as

$$\mu_B(y) = \max_x [\mu_A(x), \mu_R(x, y)] \quad (12.45)$$

This is known as *max-dot or max-product* composition.

Some examples are considered to illustrate the application of composition operation.

Composition operation can be applied between a set A and a relation R or between a relation R with another relation S. However, if a relation R is defined on X × Y and S is defined on Y × Z, we can not take intersection of R and S, because they are defined on different domains. Intersection in such cases can be taken between R and S after extending both to the domain X × Y × Z. This intersection is then projected on X × Z to find the relation between X and Z.

So the intersection of R and S = projection of $\{ce(R) \cap ce(S)\}$ on X × Z.

Example 12.15 Consider the following fuzzy relation "very near" between two sets of cities V and W, where W = [Berlin, Singapore] together with fuzzy relations R between two sets of cities U and V given by (12.37) where U = [Paris, Calcutta, Dacca] is the common set and V = [London, Calcutta].

Q = w is very near to v

		W	
		Berlin	Singapore
V	London	0.95	0.1
	Calcutta	0.15	0.9

(12.46)

We can find the relation between the sets of cities U and W by the application of max-min composition (detail steps given below).

R_1 = u is quite far from w = R o Q′ =

		W	
		Berlin	Singapore
U	Paris	0.25	0.90
	Calcutta	0.95	0.10
	Dacca	0.95	0.20

(12.47)

It is to be noted that Q′ is the transpose of Q. The relation R_1 is computed by the following steps:

Step 1.

Transposition of relation Q = Q′ =

		V	
		London	Calcutta
W	Berlin	0.95	0.15
	Singapore	0.10	0.9

Step 2. A cylindrical extension of w_1(first row) in Q′ on V × U gives relation

$$R_2 = \begin{matrix} 0.95 & 0.15 \\ 0.95 & 0.15 \\ 0.95 & 0.15 \end{matrix}$$

Step 3.

$$R_3 = \text{Intersection of } R_2 \text{ with } R = \begin{matrix} 0.25 & 0.15 \\ 0.95 & 0 \\ 0.95 & 0.15 \end{matrix}$$

Step 4.

$$\text{Projection of } R_3 \text{ on U gives the vector } w_1 = \begin{matrix} 0.25 \\ 0.95 \\ 0.95 \end{matrix}$$

Step 5. Repeating steps 2 to 4 for the vector w_2(2^{nd} row) in Q′ gives the vector w_2 (2^{nd} row) in R_1,

$$w_2 = \begin{matrix} 0.90 \\ 0.10 \\ 0.20 \end{matrix}$$

So the final relation between X and Z is given by R_1 in relation (12.47)

Instead of using max-min composition, one can compute *intersection* by any t-norm and *union* by any other t-conorm for the composition. For example using Hamacher t-norm from relation (12.20b) for intersection and algebraic sum t-conorm, s_{as} from relation (12.22a) for union the relation R_1 can be computed as

$R_1 = R \circ Q' =$

		W	
		Berlin	Calcutta
V	Paris	0.3336	0.8209
	Calcutta	0.9048	0.0995
	Dacca	0.9503	0.2616

Example 12.16 Let a set of students X = {Asoka, Alexander, Akbar} wants to take an optional subject from the Z = {fuzzy theory, fuzzy control, neural net, expert system} whose content is represented by Y = $\{y_1, y_2, y_3, y_4\}$ = {theory, application, hardware, programming}. Let the students' interest be represented by fuzzy relation R (X,Y) given by

R(X, Y) =

	y_1	y_2	y_3	y_4
Asoka	1	0.9	0.6	0.1
Alexander	0.4	0.7	1	0.2
Akbar	0.1	1	0.8	0.4

The contents of subjects are represented by fuzzy relation S(Y, Z) given by

S(Y, Z) =

	FT	FC	NN	ES
y_1	1	0.6	0.4	0.2
y_2	0.3	1	0.7	0.6
y_3	0.1	0.6	0.9	0.2
y_4	0.2	0.4	0.7	1

So, R o S =

	FT	FC	NN	ES
Asoka	1	0.9	0.7	0.6
Alexander	0.4	0.7	0.9	0.6
Akbar	0.3	1	0.8	0.6

So Asoka is advised to take Fuzzy theory, Alexander is encouraged to take Neural net and Akbar should take Fuzzy control.

12.10 EXTENSION PRINCIPLE

Let us consider the mapping $y = f(x)$ of a deterministic input variable x to an output variable y through an analytic function f. The extension principle, introduced by Zadeh, is concerned with the generalization of mapping when the input x is a fuzzy variable or fuzzy set and the function itself is deterministic or fuzzy. It provides a means of any function f that maps an n-tuple $(x_1, x_2, x_3, \ldots x_n)$ in the crisp set U to a fuzzy subsets in V. Hence, any mathematical relationship between non fuzzy elements can be extended to deal with fuzzy entities.

Given a function $f : U \to V$ and fuzzy set A in U where $A = \frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \frac{\mu_3}{x_3} + \ldots + \frac{\mu_n}{x_n}$, the extension principle states that

$$f(A) = f\left(\frac{\mu_1}{x_1} + \frac{\mu_2}{x_2} + \frac{\mu_3}{x_3} + \ldots + \frac{\mu_n}{x_n}\right) = \frac{\mu_1}{f(x_1)} + \frac{\mu_2}{f(x_2)} + \frac{\mu_3}{f(x_3)} + \ldots + \frac{\mu_n}{f(x_n)} \tag{12.48}$$

If more than one element of U is mapped to the same point y in V (that is, many to one mapping, then maximum of the membership grades is taken. The extension principle allows the function $f(x_1, x_2, x_3, \ldots x_n)$ to be extended to act on the fuzzy subsets of U, $A_1, A_2, A_3, \ldots A_n$ such that $B = f(A)$, where B is the fuzzy image of $A_1, A_2, A_3, \ldots A_n$ through $f(.)$. The fuzzy set B is defined by

$$B = \{(y, \mu_B(y)) \mid y = f(x_1, x_2, x_3, \ldots x_n), \quad (x_1, x_2, x_3, \ldots x_n) \in U\} \tag{12.49}$$

where
$$\mu_B(y) = \max_{\substack{(x_1, x_2, \ldots x_n) \in U \\ y = f(x_1, x_2, \ldots x_n)}} \min [\mu_{A_1}(x_1), \mu_{A_2}(x_2), \ldots \mu_{A_n}(x_n)] \tag{12.50}$$

with an additional condition that $\mu_B(y) = 0$ if there exists no $(x_1, x_2, x_3, \ldots x_n) \in U$ such that $y = f(x_1, x_2, x_3, \ldots x_n)$.

Example 12.17 Let U be an universe of error signal E given by U = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10] and "large error" E is a fuzzy set given by E = [0.5/6 + 0.7/7 + 0.8/8 + 0.9/9 + 1/10]. If $y = f(e) = e^2$, then by extension principle the fuzzy set B = the square of "large error" can be computed as:

$$B = \text{"large error"}^2 = 0.5/36 + 0.7/49 + 0.8/64 + 0.9/81 + 1/100$$

Example 12.18 Let A = 0.6/– 1 + 0.7/0 +1/1 + 0.5/2 be a set in the universe of discourse U = [– 2, – 1, 0, 1, 2]. The mappings on y is given by $y = f(x) = x^2$, so the membership grade $\mu_B(y)$ may be computed with extension principle. The result is shown in tabular form :

x	$\mu_A(x)$	$y = x^2$	*Max of*	$\mu_B(y)$
– 1	0.6	1	(0.6, 1)	1
0	0.7	0	(0.7)	0.7
1	1	1	(0.6, 1)	1
2	0.5	4	(0.5)	0.5

The pair of points $\{\mu_A(-1) = 0.6, \mu_A(1) = 1\}$ from A is mapped to the same point $y = 1$, so the maximum of the membership grades for these two points are to be taken for $\mu_B(y)$.

12.11 LOGICAL ARGUMENTS AND PROPOSITIONS

We first review some basic concepts and principles in classical logic and then study their generalization to fuzzy logic.

Definition 12.28 *Proposition*

Any statement that is either true or false is called a *proposition.*

All logical arguments involve *atomic propositions*, which can not be further subdivided. The atomic propositions are then combined by various *connectives* to form *compound propositions*.

There are certain propositions, called *tautologies*, that are always true. Tautologies give rise to *logical implications* and *logical equivalences*. Logical implications are basic to sound reasoning, and logical equivalences provide the means to manipulate propositions algebraically

12.11.1 Logical Arguments

As an example of logical arguments, consider the following three statements.

1. If the demand rises, then companies expand.
2. If companies expand, then they hire workers.

Chapter 12

– –

3. If the demand rises, then companies hire workers.

The logical argument has three lines, and each line contains a statement. The statements of lines 1 and 2 provide the premises of the argument, and line 3 contains the conclusion. One may or may not agree with the statement in the premises. But once the statement in the premises are accepted, the conclusion must also be accepted, since it logically follows from the premises and, therefore, the argument is sound.

Here is a second example of sound argument.

1. This computer program has a bug, or the input is erroneous.

2. The input is not erroneous.

– – – – – – – – – – – – – – – – – –

3. This computer program has a bug.

The first statement in the first example has two parts, which are statements in their own right. They are "demand rises" and "companies expand". These two statements are connected by the "if … then" construct. Similarly, the first statement in the second example consists of two parts: "this computer program has a bug and" the input is erroneous". Both parts themselves are statements, and they are connected by the word "or". To examine the correctness of arguments, Aristotle abbreviated the essential statements of the arguments by substituting letters to make it concise. The letter P may express the statement that "demand rises" the letter Q may express the statement "companies expand" and the letter R may stand for the statement "companies hire workers". Using these symbols we can express the argument involving rising demand as follows:

1. If P, then Q.

2. If Q, then R.

– – – – – – – –

3. If P, then R.

Aristotle gave a name *hypothetical syllogism* to this type of argument.

In the hypothetical syllogism, P, Q, and R each can stand for any statement. For instance, if P stands for "The cat sees the goldfish" Q for "The cat catches the goldfish" and R for "The cat eats the goldfish" then the hypothetical syllogism becomes :

1. If the cat sees the goldfish, then the cat catches the goldfish

2. If the cat catches the goldfish, then the cat eats the goldfish

– –

3. If the cat sees the goldfish, then the cat eats the goldfish

The premises of the argument may be right or wrong. The cat may be well trained and never do a thing as nasty as catching a gold fish. Moreover, even if the cat catches the goldfish, the cat may decide that cat-food is preferable to live fish. However, as soon as we accept the premises, we have no choice but to accept the conclusion, which is that cat eats the goldfish if it sees the goldfish.

In the second example, let P stand for "This computer program has a bug" and let Q stand for "the input is erroneous". Using these abbreviations, the argument can be expressed in the following way:

1. P or Q.

2. Not Q

– – – – –

3. P.

This argument is called the disjunctive syllogism and is a fundamental argument in logic.

There is one extremely important logical argument, called the modus ponens. The modus ponens can be formulated as follows:

1. If P , then Q

2. P

– – – – – – – – –

3. Q

For instance, if P is "The light turns red", and Q is "Cars stop", the premises "If the light turns red, then the cars stop", and "the light turns red "allow" one to conclude "Cars stop".

We shall call the variables P, Q and R as propositional variables. Propositional variables can only assume, in classical logic, two values, true or false. There are also two propositional constants, T and F, that represent true and false.

Definition 12.29: Compound Proposition

A proposition consisting of only single propositional variables or a single constant is called an atomic proposition. All nonatomic propositions are called compound propositions. All compound propositions contain at least one logical connective.

If the proposition is atomic, the truth-value is provided by the assignment. Otherwise, rules must be established showing how to calculate the truth-value of the compound proposition. These rules are given by the meaning of the connective. Finding the truth-value of a given assignment of a more complex proposition is more difficult. In all cases, however, a useful tool is the truth table.

In logical arguments the *"if … then"* construct is very important, which expresses the conditional relation.

Definition 12.30: Conditional Connective

Let P and Q be two propositions. Then $P \rightarrow Q$ is false if P is true and Q is false, and $P \rightarrow Q$ is true otherwise. $P \rightarrow Q$ is called the *conditional* connective of P and Q. The conditional $P \rightarrow Q$ may be translated into English by using the " if… then" construct as in "if P then Q". In other words, $P \rightarrow Q$ means that whenever P is correct, so is Q. The statement P is called the antecedent and Q the consequent.

The truth table of the conditional is given by the Table 12.2.

Table 12.2 Truth table for conditional statement

P	Q	$P \rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

From the definition of the conditional of Table 12.2, P → Q is equivalent to

$$\bar{P} \vee Q \tag{12.51}$$

and $$\bar{P} \vee (P \wedge Q) = (P \wedge Q) \vee \bar{P} \tag{12.52}$$

in the sense that they share the same truth table (Table 12.2) as P → Q, where, ¯, ∨ and ∧ represent (classical) logic operations "not," "or," and "and," respectively.

The fundamental truth for conjunction ∧, disjunction ∨, implication →, equivalence ↔ and negation ¯, are collected together in Table 12.3, where the symbols T and F denote truth and false, respectively.

New propositions, called a *logic functions*, can be defined by combination of the elementary propositions $P_1, \ldots P_n$, such that the logic function will get a particular truth value for each combination of truth values of the basic propositions. Now, n propositions can have 2^n possible combinations of truth-values, so that there will be 2^{2n} possible logic functions defining n propositions. Since, for a large n, the number 2^{2n} can be unmanageably huge in size, in classical logic the logic functions are expressed with only a few basic logic operations, called a complete *set of* primitives. The most commonly used complete set of primitives is negation ¯, conjunction, ∧ and disjunction ∨. By combining these primitives in the form of algebraic expressions-called logic formulas, new logic functions can be formed.

Table 12.3 Truth table for logic operations on elementary propositions

P	Q	$P \wedge Q$	$P \vee Q$	$P \to Q$	$P \leftrightarrow Q$	$\bar{P}$
T	T	T	T	T	T	F
T	F	F	T	F	F	F
F	T	F	T	T	F	T
F	F	F	F	T	T	T

When the proposition represented by a logic formula is always true irrespective of the truth-values of the basic propositions contained in the formula, it is called a tautology; when it is always false, it is called a contradiction.

Example 12.19 Two logic formulas, which are tautologies are given below:

$$(P \to Q) \leftrightarrow (\bar{P} \vee Q) \tag{12.53}$$

$$(P \to Q) \leftrightarrow \bar{P} \vee (P \wedge Q) \tag{12.54}$$

One way to prove the tautologies in (12.53) and (12.54), is to use the truth table method, by listing all the possible values of P and Q and see whether they are all true. The result in Table12.4 shows that (12.53) and (12.54), are tautologies.

Table 12.4 Verification of $(P \to Q) \leftrightarrow (\bar{P} \vee Q)$ and $(P \to Q) \leftrightarrow (P \wedge Q) \vee \bar{P})$

P	Q	$P \to Q$	$\bar{P} \vee Q$	$(P \wedge Q) \vee \bar{P}$		$P \to Q \leftrightarrow (\bar{P} \vee Q)$	$(P \to Q) \leftrightarrow (P \wedge Q) \vee \bar{P})$
T	T	T	T	T		T	T
T	F	F	F	F		T	T
F	T	T	T	T		T	T
F	F	T	T	T		T	T

Deductive inferences are made by using various forms of tautologies, referred to as inference rules. The three most commonly used inference rules are modus ponens, modus tollens, and hypothetical syllogism:

12.11.2 Modus Ponens

The inference rule of modus ponens states that given two propositions P and $P \to Q$ (called the premises), the truth of the proposition Q (called the conclusion) should be inferred. It is represented symbolically as

$$(P \wedge (P \to Q)) \to Q \tag{12.55}$$

An alternative intuitive representation of modus ponens is

premise 1	*x is A*
premise 2	*IF x is A THEN y is B*
Conclusion	*y is B*

12.11.3 Modus Tollens

The inference rule of modus tollens states that, given two propositions $\overline{Q}$ and $P \to Q$ the truth of the proposition $\overline{P}$ should be inferred. It is represented symbolically as

$$(\overline{Q} \wedge (P \to Q)) \to \overline{P} \tag{12.56}$$

The intuitive representation of modus tollens is

premise 1	*y is not B*
premise 2	*IF x is A THEN y is B*
Conclusion	*x is* not A

12.11.4 Hypothetical Syllogism

The inference rule of hypothetical syllogism states that given two propositions $P \to Q$ and $Q \to R$, the truth of the proposition $P \to R$ should be inferred. Symbolically, it becomes

$$\{(P \to Q) \wedge (Q \to R)\} \to (P \to R) \tag{12.57}$$

The alternative representation of hypothetical syllogism is

premise 1 :	*IF x is A THEN y is B*
premise 2 :	*IF* y is *B THEN z is C*
Conclusion :	*IF x is A THEN z* is *C*

12.12 INTERPRETATIONS OF FUZZY IF-THEN RULES

Because the fuzzy propositions are interpreted as fuzzy relations, the key question remaining is how to interpret the IF-THEN operation. Because fuzzy IF-THEN rules can be viewed as replacing the P and Q with fuzzy propositions, we can interpret the fuzzy IF-THEN rules by replacing the, $^{-}$, $\vee$ and $\wedge$ operators in (12.16) (12.14) and (12.15) with fuzzy complement, fuzzy union, and fuzzy intersection, respectively. Since there are a wide variety of fuzzy complements, fuzzy union, and fuzzy intersection operators, a number of different interpretations of fuzzy IF-THEN rules were proposed in the literature. We list here only Mamdani [44] and Larsen [45] implication which are widely used in Fuzzy Control.

In the following, we replace the P and Q in (12.51) and (12.52) by FP_1 and FP_2, respectively, where FP_1 and FP_2 are fuzzy propositions. We assume that FP_1, is a fuzzy relation defined in $U = U_1 \times \ldots \times U_n$, FP_2 is a fuzzy relation defined in $V = V_1, \times \ldots \times V_m$, and x and y are linguistic variables (vectors) in U and V, respectively.

Chapter 12

Conceptually, we can replace the $^-$, $\vee$ and $\wedge$ in (12.51) and (12.52) by any fuzzy complement, *t*-conorm and *t*-norm, respectively, to obtain a particular interpretation. However, the guidelines for making a particular choice is beyond the scope of the present book. When P and Q are crisp propositions $P \rightarrow Q$ is a global implication in the sense that Table 12.2 covers all the possible cases. However, when P and Q are fuzzy propositions, $P \rightarrow Q$ may only be a local implication in the sense that $P \rightarrow Q$ has large truth value only when both P and Q have large truth values. Therefore, the fuzzy IF-THEN rule

$$IF < FP_1 > THEN < FP_2 > \tag{12.58}$$

should be interpreted as

$$IF < FP_1 > THEN < FP_2 > ELSE < NOTHING > \tag{12.59}$$

where *NOTHING* means that this rule does not exist. In terms of logical equivalence it becomes

$$P \rightarrow Q = P \wedge Q \tag{12.60}$$

Using min or algebraic product for the $\wedge$ in (12.60), we obtain the Mamdani or Larsen product implications, which are widely used in control systems.

(a) Mamdani Implications (Min operation): The fuzzy IF-THEN rule (12.58) is interpreted as a fuzzy relation R_M in U × V with the membership function

$$\mu_M(x, y) = \min [\mu_{FP1}(x), \mu_{FP2}(y)] \tag{12.61}$$

(b) Larsen Implications (Product operation): The implication of (12.58) is interpreted as a fuzzy relation R_P in U × V with the membership function

$$\mu_P(x,y) = \mu_{FP1}(x)\, \mu_{FP2}(y) \tag{12.62}$$

In general, we shall designate the relation implied by fuzzy rules of the form given in Equation (12.58) in future as follows

$$R(x, y) = \text{Imp}\, (FP_1(x), FP_2(y)) = FP_1(x) \rightarrow FP_2(y)$$

where Imp stands for any implications from equations (12.61)-(12.62)

Mamdani implications are the most widely used implications in fuzzy systems, especially in fuzzy control systems. They are supported by the argument that fuzzy IF-THEN rules are purely local.

Let us now consider some examples for the computation of the Mamdani implications relation R_M.

Example 12.20. Let U = {220, 200, 180, 160, 140} and V = {1000, 900, 800, 700}. Suppose that the voltage x, $x \in$ U applied to the dc motor develops a speed y, $y \in$ V. Also suppose that voltage and speed are related by the following fuzzy *if-then* rule:

$$\text{IF } x \text{ is about 220 volts, THEN } y \text{ is about 800 rpm} \tag{12.63}$$

where the fuzzy sets "about 220 volts" and "about 800 rpm" are defined as

$$\text{about 220 volts} = A = 1/220 + 0.75/200 + 0.5/180 + 0.25/160 + 0/140 \tag{12.64}$$

$$\text{about 800 rpm} = B = 0.3/1000 + 0.6/900 + 1/800 + 0.6/700 \tag{12.65}$$

Mamdani Implication: Relation, R_M is found as a *max-min composition* of sets A and B in X × Y written as R_M = A *o* B and can be computed by following the procedure of Section 12.9.4. Alternatively it can be computed by operation similar to matrix multiplication of two vectors xy', where x' = [1 0.75 0.5 0.25 0] and y' = [0.3 0.6 1 0.6], (here x' stands for transpose of the vector x) with product of matrix elements replaced by *minimum* and sum is replaced by or (*maximum*) operation. The final result for R_M for the above example is given below :

			y			
			1000	900	800	700
		220	0.3	0.6	1	0.6
R_M =	x	200	0.3	0.6	0.75	0.6
		180	0.3	0.5	0.5	0.5
		160	0.25	0.25	0.25	0.25
		140	0	0	0	0

(12.66)

Note that the element R_M (1, 1) is the minimum of (1, 0.3) = 0.3, R_M (1, 3) is the minimum of (1, 1) = 1 and so on.

12.12.1 Fuzzy Relation Equations

Fuzzy relation equations play an important role in areas such as fuzzy system analysis, design of fuzzy controllers, decision-making processes, and fuzzy pattern recognition.

The notion of fuzzy relation equations is associated with the concept of composition of fuzzy relation. Here we shall focus on the max-min composition because it has been studied extensively and has been utilized in numerous applications. Let A be a fuzzy set in X and R (X, Y) be a binary fuzzy relation in X × Y. The set-relation composition of A and R, A *o* R, results in a fuzzy set in Y. Let us denote the resulting fuzzy set as B. Then we have

$$A \, o \, R = B, \tag{12.67}$$

whose membership function is

$$\mu_B(y) = \mu_{A\,o\,R}(y) = \max_{x \in X} \min[\mu_A(x), \mu_R(x, y)] \tag{12.68}$$

Equation (12.67) is a so-called fuzzy relation equation. If we view R as a fuzzy system with A as a fuzzy input and B as a fuzzy output, then we can consider Equation (12.67) as describing the characteristics of a fuzzy system via its fuzzy input-output relation as shown in Figure 12.13. Hence, given a fuzzy input A to a fuzzy system R, a fuzzy output B can be computed by Equation (12.67). Modus ponen logic is easier with fuzzy relation equation, but not the modus tollen logic. When A and R are given, the fuzzy set B can be easily computed from Equation (12.68). However, it can be shown that when A and B are given, the relation R in Equation (12.67) can be determined only if the following condition holds:

$$\max_{x \in X} \mu_A(x) \geq \mu_B(y) \quad \text{for all } y \in Y \tag{12.69}$$

Similarly, it can be shown that when B and R are given, the equation (12.67) has no solution if the following inequality holds :

$$\max_{x \in X} \mu_R(x, y) < \mu_B(y) \qquad \text{for some } y \in Y \tag{12.70}$$

Example 12.21. Given a fuzzy set A in X and a fuzzy relation R in X x Y as follows,

$$A = 0.2/x_1 + 0.8/x_2 + 1/x_3 = \begin{matrix} x_1 & x_2 & x_3 \\ (0.2 & 0.8 & 1) \end{matrix}$$

$$R = \begin{matrix} & y_1 & y_2 & y_3 \\ x_1 \\ x_2 \\ x_3 \end{matrix}\begin{bmatrix} 0.7 & 1 & 0.4 \\ 0.5 & 0.9 & 0.6 \\ 0.2 & 0.6 & 0.3 \end{bmatrix}$$

we then have $$B = \overset{\begin{matrix} x_1 & x_2 & x_3 \end{matrix}}{(0.2 \quad 0.8 \quad 1)} \cdot \begin{matrix} & \begin{matrix} y_1 & y_2 & y_3 \end{matrix} \\ \begin{matrix} x_1 \\ x_2 \\ x_3 \end{matrix} & \begin{bmatrix} 0.7 & 1 & 0.4 \\ 0.5 & 0.9 & 0.6 \\ 0.2 & 0.6 & 0.3 \end{bmatrix} \end{matrix} = \overset{\begin{matrix} y_1 & y_2 & y_3 \end{matrix}}{(0.5 \quad 0.8 \quad 0.6)}$$

$$B = 0.5/y_1 + 0.8/y_2 + 0.6/y_3$$

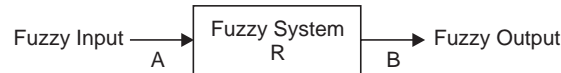



Fig. 12.13 Fuzzy system with fuzzy input and fuzzy output

The fuzzy relation R used in (12.67) is usually not supplied directly but is embedded even in a single conditional fuzzy proposition. The relation R that is embedded in a conditional fuzzy proposition p of the form :

$$p : \text{if } x \text{ is A then } y \text{ is B}$$

is computed from for $x \in X$ and $y \in Y$ by the formula

$$R(x, y) = \text{Imp}\ [A(x), B(y)] \tag{12.71}$$

where Imp stands for any of the implications including those in relations (12.61) and (12.62)

12.13 BASIC PRINCIPLES OF APPROXIMATE REASONING

The ultimate goal of fuzzy logic is to provide foundations for approximate reasoning with imprecise propositions using fuzzy set theory as the principal tool. In order to achieve this goal, the *generalized modus ponens, generalized modus tollens,* and *generalized hypothetical syllogism* were proposed, which are the fundamental principles in fuzzy logic. These generalization are based on compositional rule of inference.

12.13.1 Generalized Modus Ponens

The modus ponens of the classical logic is generalized to get fuzzy inference rules as follows

premise 1 : *x is A′*

premise 2 : *IF x is A THEN y is B,*

Conclusion : *y is B′*

For fuzzy sets A, A′, B and B′ and given fuzzy propositions 1 and 2, we infer a new fuzzy proposition for the conclusion such that closer the set A′ to A, the closer the set B′ to B.

The set B′ is computed by using the relations (12.67) and (12.71).

12.13.2 Generalized Modus Tollens

The classical Modus Tollens of (12.36) can be framed for fuzzy inference rules as follows:

premise 1 : *y is B′*

premise 2 : *IF x is A THEN y is B*

conclusion : *x is A′*

For fuzzy sets A, A′, B and B′ and given fuzzy propositions 1 and 2, we infer a new fuzzy proposition for the conclusion such that farther the set A′ is from A, farther is the set B′ from B.

The set A′ can be found such that the relation (12.72) given below is satisfied for a given set B′.

$$B' = A' \circ R \tag{12.72}$$

where R is computed from (12.71). For the existence of the solution (12.72), $\mu_B(y)$ must satisfy the necessary condition

$$\max_{x \in X} \mu_R(x, y) > \mu_B(y) \qquad \text{for some } y \in Y \tag{12.73}$$

However, since the condition is only necessary, compliance of the constraint in relation (12.73) may not always lead to a solution.

12.13.4 Generalized Hypothetical Syllogism

We now take up the extension of the classical Hypothetical Syllogism of (12.57) to generalized fuzzy hypothetical syllogism based on two conditional fuzzy production rules. This may be stated as follows.

Premise 1 : *IF x is A THEN y is B*

Premise 2 : *IF y is B′ THEN z is C*

Conclusion : *IF x is A THEN z is C′*

For the fuzzy sets A, *B, B′, C* and C′ and fuzzy propositions 1 and 2, the new fuzzy proposition in the conclusion is inferred such that closer the set B to B′, the closer the set C′ to C.

For each of the conditional fuzzy propositions, we can find fuzzy relations R_1, R_2 and R_3 by using (12.71) as follows.

$$R_1(x, y) = \text{Imp } [A(x), B(y)],$$
$$R_2(y, z) = \text{Imp } [B(y), C(z)],$$
$$R_3(x, z) = \text{Imp } [A(x), C(z)]$$

The compositional relation may also be written in matrix form

$$R_3 = R_1 \, o \, R_2 \tag{12.74}$$

Example 12.22. Consider the relational matrices between two sets of cities (U, V) and (V, W) through fuzzy relations given by equations (12.37) and (12.46) in Section 12.9.4 which are renamed as R_1 and R_2 and reproduced below:

$$R_1 = \begin{bmatrix} 0.25 & 0.90 \\ 0.95 & 0.0 \\ 1.0 & 0.20 \end{bmatrix}, \quad R_2 = \begin{bmatrix} 0.95 & 0.1 \\ 0.15 & 0.9 \end{bmatrix}$$

The fuzzy relational matrix between the sets of cities U and W is then found by computing R_3 from equation (12.74) as:

$$R_3 = \begin{bmatrix} 0.25 & 0.90 \\ 0.95 & 0.10 \\ 0.95 & 0.20 \end{bmatrix}$$

12.14 REPRESENTATION OF A SET OF RULES

Suppose the error signal e and its derivative $\dot{e}$ be two inputs to a fuzzy controller to produce controller output u. This is symbolically represented as

if e is PB and $\dot{e}$ is NB **then** u is PS

which stands for the statement in natural language

if the current value of the error signal, e is ***positive big*** *and the current value of the rate of change of error,* $\dot{e}$ *is* ***negative big*** *th*en the output of the controller should be set to a value of ***positive small.***

Alternatively

*If the property of the current value of the error signal, e is **positive big** and it is the case that the property of the current value of the rate of change of error,* $\dot{e}$ *is **negative big** then it is proper to set the property of the controller output, u to **positive small.***

Assuming that the controller characteristic is described by the functional relationship $y = f(x)$ as shown in Figure 12.14, the functional relationship can then be approximated by the following set of if-then rules :

if x is Z then y is Z

if x is S then y is S

if x is M then y is M

if x is B then y is B

where Z stands for around zero value, S stands for small value, M stands for medium value and B stands for big value.

The result of approximation is shown in Figure 12.14.The production rules explicitly state that the value of Y can be computed when X is known.

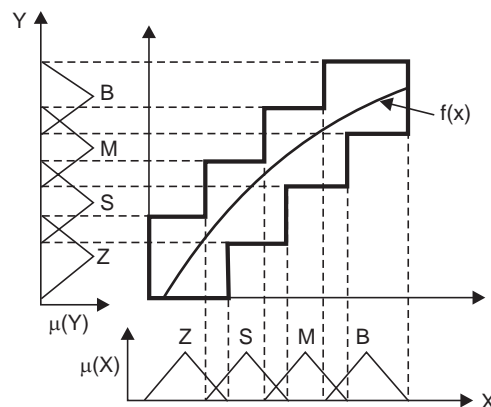



Fig. 12.14 Fuzzy set approximation of $y = f(x)$

Since x is the cause and y is the effect, and signal flows from x to y, there is no rule for finding the value of x when y is known.

Before we take up multiple production rules involving more than one state in a fuzzy control system, let us illustrate graphically the computation of the membership of the controller output u from the following single if-then rule

Rule 1 : If the error signal e is NB then the controller output, u is PS where both e and u are assumed to have 7 membership functions to cover the entire normalized domains – 10 to + 10, in E and U respectively. We assume the membership functions to be triangular in shape as shown in Figure 12.15, where ZE stands for zero while PS, PM, PB, NS, NM, NB stand for positive small, positive medium, positive big, negative small, negative medium, negative big respectively are the term sets (vide Section 12.8.1 for details)

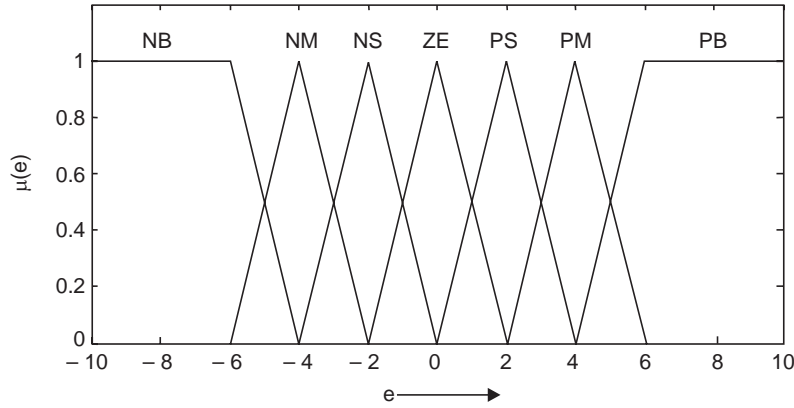



Fig. 12.15 Membership functions for e and u

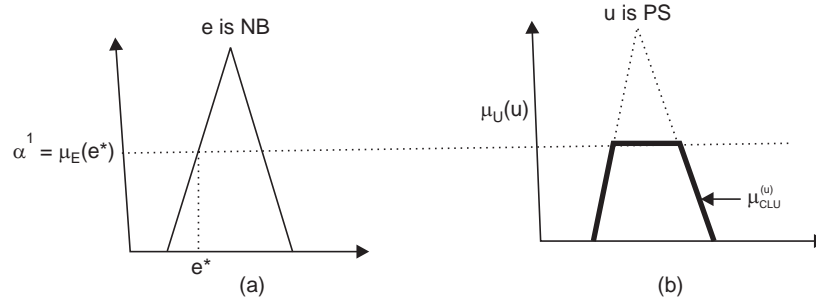



Fig. 12.16 Interpretation of Mamdani implication

The membership functions of the controller output u in PS may be obtained for the above rule by using equation (12.68), where R is computed from any implications including those given by the relations (12.61)-(12.62). So we can write

$$\mu_{PS}(u) = \max_{e \in NB} \min[\mu_E(e), \mu_R(e, u)] \text{ for all } u \in PS \quad (12.75)$$

Using Mamdani implication from equation (12.61), we find $\mu_R(e, u) = \min [\mu_E(e), \mu_U(u)]$

In order to compute, $\mu_U(u)$, we refer to Figure 12.16a and note that at a particular crisp value of $e = e^*$, $\mu_E(e^*) = \alpha^1$ is a fixed value in NB. So $\mu_U(u)$, in turn, is found using (12.61) as

$$\mu_U(u) = \min[\alpha^1, \mu_{PS}(u)] \text{ for all } u \in PS \quad (12.76)$$

$$= \mu_{CLU}(u) \quad (12.77)$$

where $\mu_{CLU}(u)$ is the membership of u (dotted triangle in Figure 12.16(b)) in PS clipped by α^1.

The clipped membership function of u in PS is shown in solid line in Figure 12.16 (b). So, for a single rule, $\mu_{PS}(u)$ is given by $\mu_{CLU}(u)$ in relation (12.77).

The value of $\mu_E(e^*) = \alpha^1$ is the value of the membership function $\mu_E(e)$ at a particular value of the error and is known as *degree of consistency* between the given fact e^* and the antecedent of Rule 1 given above. In general, if there exists a $\mu_{CLU}(u) > 0$ for $\alpha^1 > 0$ then the Rule is said to have fired with a firing strength α^1.

12.14.1 Approximate Reasoning with Multiple Conditional Rules

We shall now take up the computation of the membership functions of controller outputs u from multiple conditional rules.

Typical multiple conditional rules are of the form :

$$\begin{array}{ll} \text{Rule 1 :} & \text{if } x \text{ is } A_1 \text{ then } y \text{ is } B_1 \\ \text{Rule 2 :} & \text{if } x \text{ is } A_2 \text{ then } y \text{ is } B_2 \\ \text{Rule 3 :} & \text{if } x \text{ is } A_3 \text{ then } y \text{ is } B_3 \\ \text{Rule } n \text{ :} & \text{if } x \text{ is } A_n \text{ then } y \text{ is } B_n \\ \text{Fact :} & x \text{ is } A' \\ \hline \text{Conclusion : } y \text{ is } B' & \end{array} \tag{12.78}$$

Given the n fuzzy if-then rules in the form (12.78), and a fact "x is A′", one concludes "y is B′", where $A', A_j \in f(x), B', B_j \in f(y)$ for all $j \in N_n$ and x and y are sets of values in X and Y. The set B′ in (12.78) is computed by a method known as interpolation which is a special case of compositional rule of inference. The step by step procedure for computation of the set B′ is given below:

Step 1. Calculate the degree of consistency, α^j, between the given fact A′ and the antecedent of each if-then rule j in terms of **height** of the intersection of the associated sets A′ and A_j .

$$\alpha^j = h(A' \cap A_j) = \max_{x \in X_n} \min [A'(x), A_j(x)] \tag{12.79}$$

Step 2 Calculate the conclusion B′ by clipping each set B_j by the value α^j, which gives the degree to which the antecedent A_j is consistence with the given fact A′ and taking the unions of the clipped sets.

$$B'(y) = \max_{j \in N_n} \min [\alpha_j, B_j(y)] \text{ for all } y \in Y \tag{12.80}$$

This is illustrated in Figure 12.17 for two rules.

In order to show that the interpolation is a special case of compositional rule of inference, we consider the fuzzy relation R on X × Y defined as

$$R(x, y) = \max_{j \in N_n} \min [A_j(x), B_j(y)] \text{ for all } x \in X, y \in Y \tag{12.81}$$

It can be demonstrated [74] by using (12.61) and (12.80) that the set B′ obtained by (12.81) is equivalent to that obtained by the max-min composition of the set A′ with the relation R, that is

$$B' = A' \, o \, R \tag{12.82}$$

where o denotes the max-min composition.

We can write,

$$\begin{aligned} B'(y) &= \max_{j \in N_n} \min [\alpha^j, B_j(y)] \\ &= \max_{j \in N_n} \min [\max_{x \in X} A'(x), A_j(x), B_j(y)] \\ &= \max_{j \in N_n} \max_{x \in X} [\min (A'(x), A_j(x), B_j(y))] \\ &= \max_{j \in N_n} \max_{x \in X} \min [A'(x), (A_j(x), B_j(y))] \\ &= \max_{x \in X} \min [A'(x), \max_{j \in N_n} (A_j(x), B_j(y))] \\ &= \max_{j \in N_n} \min [A'(x), R(x, y)] \\ &= (A' \, o \, R)(y) \end{aligned}$$

Hence, the result $B' = A' \; o \; R$

It is to be observed that the fuzzy relation R employed in the reasoning is obtained from the if-then rules in (12.78) by Mamdani implications. That is

$$R_j(x, y) = \min [A_1(x), B_1(y)] \text{ for all } x \in X, y \in Y \quad (12.83)$$

Then R is found by the unions of relations R_j for all rules in (12.78)

$$R = \bigcup_{j \in N_n} R_j \quad (12.84)$$

The basis for arriving at the relation above, is the assumption that the fuzzy *if-then rules* are **disjunctive** so that the conclusion is computed for at least one j for a given set A′ whenever $\alpha^j(A') > 0$. A rule j is said to have fired for a given fact A′ if $\alpha^j(A') > 0$.

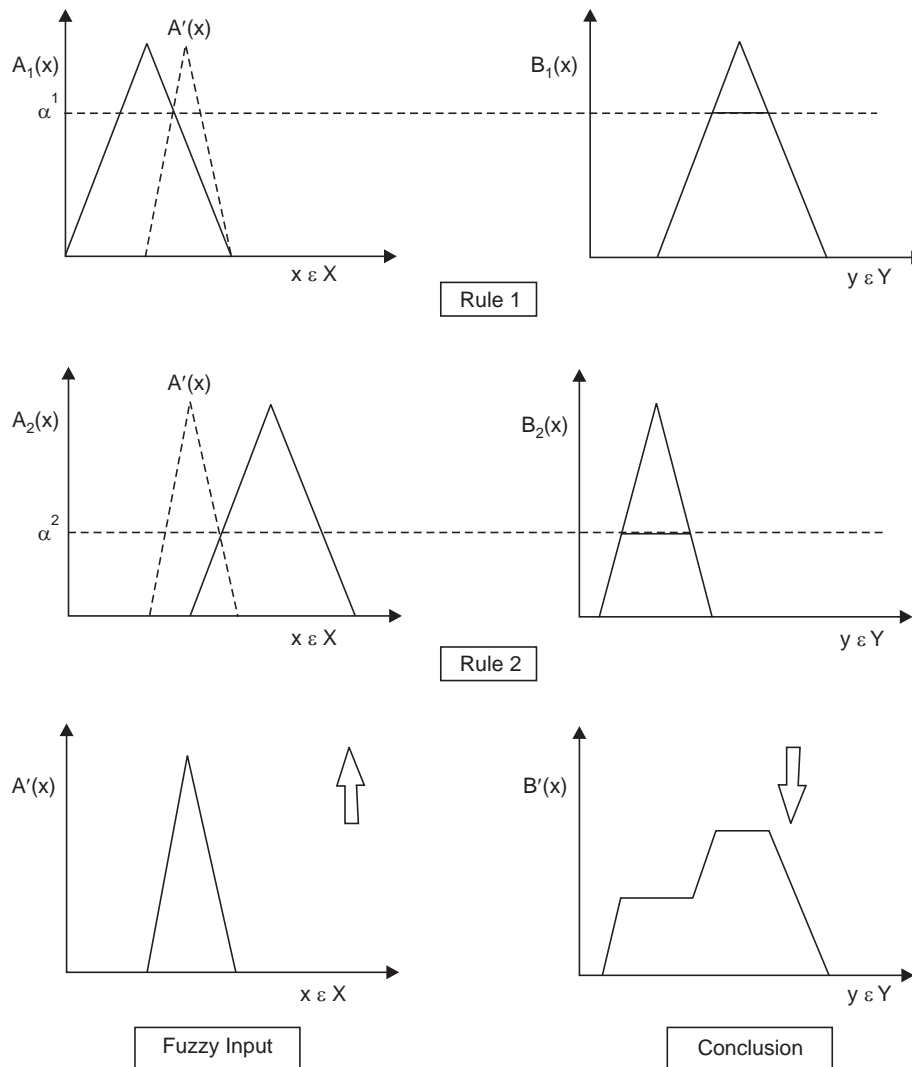



Fig. 12.17 Graphical illustration of interpolation method

MATLAB SCRIPTS

```
%Script_Fig12.3
% Fuzzy representation of 'Young', 'Middle Aged' and 'Old' as a
% function of age
x = (0:0.5:100)′;
y1 = gaussmf(x, [20 50]); y2 = gaussmf(x, [20 0]); y3 = gaussmf(x, [20 100]);
plot(x, [y1 y2 y3]);
axis=([0 100 0 1.2]);
xlabel('x=Age'); ylabel('\mu(x)');
```

```
%Script_Fig12.8
% Sugeno Complement
clear all ; close all hidden
x=(0:0.01:1)′; LL=length(x);
L=[-0.9 0 1 10];
for j=1:4
for i=1:LL   % Refer to equation (12.18)
num1=(1-x(i)); den1=(1+L(j)*x(i));
y=num1/den1;Ys(:,i)=y;
end
plot(x,Ys); hold on
axis([0 1.1 0 1.2]);
xlabel('x'),ylabel('Sugeno Complement')
end
```

```
%Script_Fig12.11
%Representation of car speed using trapeziodal MF
% The trapezoidal MF reduces to triangular MF with b=c in relation (12.26)
clear all; close all hidden
x = (0: 1:100)′;
y1 = trapmf(x, [-10 0 35 55]);
y2 = trapmf(x, [35 55 55 75]);                    % b=c
y3 = trapmf(x, [55 75 100 120]);
plot(x, [y1 y2 y3]);
xlabel('x=Speed of car , Kmph'); ylabel('\mu(x)');
%Use axis properties under drop down menu edit in figure window to
%set x axis limits of 0 to 1.2 and y axis limits of 0 to 100 for better view
```

```
%Script_Fig12.12
% Plot of Figure 12.12
x = (-10:0.1:10)';
y1 = trapmf(x, [-20 -10 -6 -4]); y2 = trapmf(x, [-6 -4 -4 -2]);
y3 = trapmf(x, [-4 -2 -2 0]); y4 = trapmf(x, [-2 0 0 2]);
y5 = trapmf(x, [0 2 2 4]); y6 = trapmf(x, [2 4 4 6]);
y7 = trapmf(x, [4 6 10 20]);
plot(x, [y1 y2 y3 y4 y5 y6 y7]);
axis([-10 10 0 1.1])
xlabel('x \rightarrow'); ylabel('\mu(x) \rightarrow');
```

PROBLEMS

P12.1 Determine intuitive membership functions for "short persons" and "tall persons"

P12.2 Model the following expressions as fuzzy sets: (*a*) hard-working students, (*b*) top students, and (*c*) average students.

P12.3 Given the fuzzy sets A, B and C defined in the interval U = [0, 10] by the membership functions

$$\mu_A(x) = \frac{x}{x+2}, \quad \mu_B(x) = 2^{-x}, \; \mu_C(x) = \frac{1}{1+5(x-2)^2}$$

Now find the mathematical expressions for each of the following fuzzy sets and plot their membership functions:

(*a*) $\overline{A}, \overline{B}, \overline{C}$ (*b*) $A \cup B, B \cup C, A \cup C$

(*c*) $A \cap B, B \cap C, A \cap C$ (*d*) $A \cup B \cup C, A \cap B \cap C$

(*e*) $A \cap \vec{C}, \overline{\overline{B} \cap C}, \overline{A \cup C}$

P12.4 Let $A(x) = \begin{cases} x & \text{if } 0 \le x \le 1 \\ 2-x & \text{if } 1 \le x \le 2 \\ 0 & \text{otherwise} \end{cases}$ and $B(x) = \begin{cases} \dfrac{2x-1}{2} & \text{if } \dfrac{1}{2} \le x \le \dfrac{3}{2} \\ \dfrac{6-2x}{3} & \text{if } \dfrac{3}{2} \le x \le 3 \\ 0 & \text{otherwise} \end{cases}$

where $A'(x) = 1 - A(x)$ and $B'(x) = 1 - B(x)$ for $x \in [0, 3]$.

Show that $A(x) \wedge A'(x) = (A(x) \wedge A'(x) \wedge B(x)) \vee (A(x) \wedge A'(x) \wedge B'(x))$

P12.5 For $x \in [0, 1]$, let $A(x) = \sin \pi x$, $B(x) = x^2$, $A'(x) = 1 - A(x)$, and $B'(x) = 1 - B(x)$,

Show that $A(x) \wedge A'(x) = (A(x) \wedge A'(x) \wedge B(x)) \vee (A(x) \wedge A'(x) \wedge B'(x))$

Also find the algebraic expressions for all the functions involved and obtain their graphical plot.

P12.6 Show that the function $f(x) = \max \left\{\alpha \min\left(\dfrac{x-a}{b-a}, 1, \dfrac{d-x}{d-c}\right), 0\right\}$

yields the trapezoidal set determined by the points $(a, 0), (b, \alpha), (c, \alpha), (d, 0)$ for $a < b < c < d$ and $0 < \alpha \le 1$.

P12.7 Let $\mu_A(x) = A[a, b, c]$ denote the triangular membership function determined by the points $(a, 0)$, $(b, 1)$, $(c, 0)$.

(*i*) Plot the fuzzy sets $A_1[2, 4, 6]$, $A_2[4, 6, 8]$, $B_1[1, 4, 6]$, $B_2[4, 8, 9]$, C_1 [1, 4, 7] and C_2 [3, 6, 9].

(*ii*) Also plot the aggregated fuzzy set realized by the Mamdani method

$$u(3, 3)(z) = (\mu_{A1}(3) \wedge \mu_{B1}(3) \wedge \mu_{C1}(z)) \vee (\mu_{A2}(3) \wedge \mu_{B2}(3) \wedge \mu_{C2}(z))$$

and defuzzify $u(3, 3)(z)$ by the "center average" method.

P12.8 With fuzzy sets A and B defined as in 12.3

(*a*) Determine the membership functions for $A \cup B$ and $A \cap B$ using the Einstein *s*-norm (12.22*b*) and *t*-norm (12.20*d*) with $c = 2$.

(*b*) Using (12.16) as fuzzy complement, algebraic sum (12.22*a*) as fuzzy union, and algebraic product (12.20*a*) as fuzzy intersection, compute the membership functions for $A \cap B$, $A \cup B$, and $\overline{A} \cup B$

P12.9 Consider the three binary fuzzy relations defined by the relational matrices:

$$Q_1 = \begin{pmatrix} 1 & 0 & 0.8 \\ 0.4 & 0.3 & 0 \\ 0 & 0.6 & 1 \end{pmatrix}, \quad Q_2 = \begin{pmatrix} 0.5 & 0.7 & 0 \\ 0 & 0.8 & 0.2 \\ 0 & 0.3 & 0 \end{pmatrix}, \quad Q_3 = \begin{pmatrix} 1 & 0 & 0.8 \\ 0 & 1 & 0 \\ 0.8 & 0 & 1 \end{pmatrix}$$

Compute the max-min and max-product composition $Q_1 \; o \; Q_2$, $Q_1 \; o \; Q_3$ and $Q_1 \; o \; Q_2 \; o \; Q_3$

P12.10 For the fuzzy set $A = 0.6/-1 + 0.8/0 + 1/1 + 0.4/2$ and the function $f(x) = x^3$, compute the fuzzy set $f(A)$ using the extension principle.

P12.11 Let $U = \{x_1, x_2, x_3\}$ and $V = \{y_1, y_2\}$, and assume that a fuzzy IF-THEN rule "IF x is A, THEN y is B" is given,

where $A = 0.4/x_1 + 1/x_2 + 0.6/x_3$ and $B = 1/y_1 + 0.5/y_2$. Then, given a fact "x is A′", where $A' = 0.5/x_1 + 0.8/x_2 + 0.6/x_3$, use the generalized modus ponens to derive a conclusion in the form "y is B′", where the fuzzy relation $A \to B$ is interpreted using :

(a) Mamdani implication (12.61).

(b) Larsen implication (12.62)

P12.12 Assuming that U, V, A and B are the same as in P12.11 and given a fact "y is B′", where $B' = 0.8/y_1 + 0.6/y_2$, derive a conclusion "x is A′" using the generalized modus tollens of relation where the fuzzy relation $A \to B$ is interpreted using:

(a) Larsen implication (12.62)

(b) Mamdani implication (12.61)

CHAPTER

13

Fuzzy Logic Based Controller

13.1 THE STRUCTURE OF FUZZY LOGIC-BASED CONTROLLER

The fuzzy logic-based controller (or Fuzzy Controller) was first implemented by Mamdani and Assilian [46] based on the fuzzy logic system generalized from fuzzy set theory introduced by Zadeh [1]. The block diagram of a fuzzy logic controller (FLC) is shown in Fig. 13.1 consisting of the following main units:

1. a **fuzzifier** module
2. a **fuzzy inference engine**
3. the **knowledge base** containing a rule base and database
4. a **defuzzifier module**

In fuzzy control applications, the observed data are usually crisp. So, fuzzification is necessary at the input stage of the FLC, for data manipulation by fuzzy set theory. Fuzzification translates the input crisp data into the fuzzy representation incorporating the vagueness and imprecision in a natural language, for further processing in FLC. The most outstanding feature of fuzzy set theory, which made it very attractive for applications, is its ability to model the meaning of natural language expressions.

A fuzzy system is modeled by a set of linguistic conditional statements prepared by an expert that is usually represented in the form of **"if-then"** rules as

if (a set of conditions are satisfied) **then** (a set of consequences can be inferred) (13.1)

The antecedent and the consequence of these **if-then** rules are associated with fuzzy concepts, so they are often called **fuzzy conditional statements**. In fact, the antecedent is a condition in its application domain and the consequence is a control action for the system under control. Above all, the fuzzy control rules provide a convenient way for expressing control policy and domain knowledge. The knowledge base module is used to specify the control rules, which comprises knowledge of the application domain and the attendant control goals. Moreover, to deal with the fuzzy information described above, the fuzzy inference engine employs the fuzzy knowledge base to simulate human decision-making and infer fuzzy control actions [47]. Finally, the defuzzifier module is used to translate the processed fuzzy data into the crisp data suited to real world applications.

Since rules specify the implication relationship between the input variables and output variables characterized by their corresponding membership functions, the choice of the rules along with the membership functions makes significant impacts on the final performance of the fuzzy logic controller and therefore becomes a major strategy in fuzzy logic controller design.

The more membership functions are used, the more the rules emerge, and the finer the results of inference, but this increases the computational complexity.

A fuzzy logic controller designed on the basis of the fuzzy logic is an approximate reasoning-based controller, which does not require exactly analytical models and is much closer in spirit to human thinking and natural language than the traditional logic system.

Essentially, the control strategies in the FLC are based on expert experience, so the fuzzy logic controller can be regarded as the simulation of a humanoid control model. When designing a FLC, the control strategies have to be based on the determination of the fuzzy membership function of control variables and the linguistic control rules pre-constructed by experts to accomplice inference. Therefore, after finishing the design of a controller, if the control result fails to meet the system requirements due to a change in the outside environment of the control system, the system control strategies have to be modified to fit the control objective. The possible solution to this problem is that we can adjust either the membership function of the fuzzy sets or the control rules to achieve the control objective.

Let us examine the principal blocks of the FLC more closely.

13.1.1 Knowledge Base

The knowledge base of the fuzzy controller consists of **database and rule base**. The basic function of the rule base is to provide the necessary information for proper functioning of the fuzzification module, the rule base, and the defuzzification module. It contains information about

(*i*) *the membership functions (fuzzy sets) representing the meaning of the linguistic values*

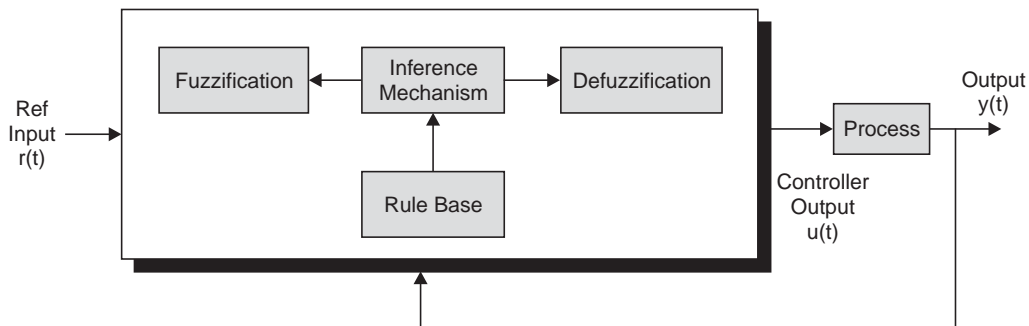
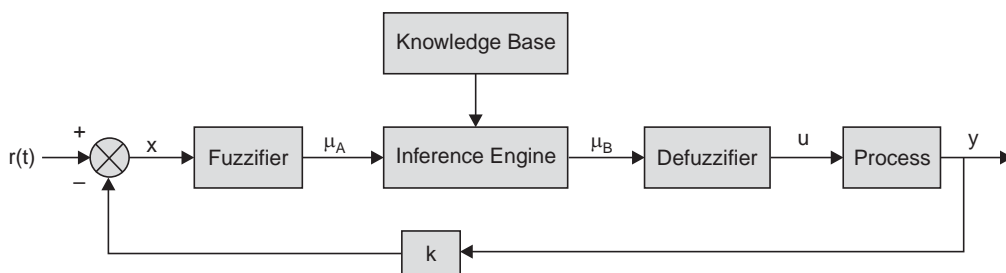



Fig. 13.1 Basic fuzzy controller (*a*) structure (*b*) detailed block diagram

(*ii*) *the physical domains and their normalized counterpart together with scaling factors for normalization and denormalization*

(*iii*) *the rule base about the control strategy of an experienced process operator or control engineer in the form of production rules such as* ***if*** *(the property of system state)* ***then*** *(the property of controller output)*

The **if** part of the production rules contains the linguistic description of a process state in terms of a logical combination of atomic fuzzy proposition.

Then part of the production rules contain the description of the controller output in terms of a logical combination of fuzzy propositions concerning the values of the controller output whenever the process states in the antecedent matches to certain degree.

The construction of production rules concerns the choice of the following:

(*a*) system state and controller output variables

(*b*) the contents of the rule antecedent and the rule-consequent

(*c*) the term set of the linguistic values for the system state and controller output variables

However, the most important step for the design of controller is the derivation of the set of rules for system under consideration.

13.1.2 Rule Base

13.1.2.1 Choice of Sate Variables and Controller Variables

When the system states are available for measurement and control, the rule base are written in terms of the state variables instead of error and its derivatives. In such a case the production rules for multi input single output are:

if x_1 *is (linguistic term) and* x_2 *is (linguistic term) and* x_n *is (linguistic term) then* u *is (linguistic term)*

Rules of this type are derived from fuzzy process model. Another distinct rule, popular in fuzzy control system, may be stated as:

If x_1 is (linguistic term) and x_2 is (linguistic term) ... and x_n is (linguistic term) then

$$u = f(x_1, x_2 \ldots x_n)$$

where f is a function of the state variables of the system x_i ($i = 1, 2 \ldots n$).

This type of controller is referred to as **Sugeno type** of controller [48-49].

We consider a multi-input-single-output (MISO) fuzzy system where the input space $U = U_1 \times U_2 \times \ldots \times U_n, \subset R^n$ and the output space $V \subset R$. *We consider only the multi-input-single-output case, because a multi-output system can always be decomposed into a collection of single-output systems.* For example, if we are asked to design a 5-input-3-output fuzzy system, we can first design three 5-input-1-output fuzzy systems separately and then put them together.

In general, let x be the system state vector, y is the system output and u is the controller output, which is input to the process.

The general form of the fuzzy control rules S^r in the case of multi-input-single-output systems (MISO) can, therefore, be written as:

$$S^r : \text{if } x_1 \text{ is } A_1^{\,r} \text{ and } x_2 \text{ is } A_2^{\,r} \ldots, x_n \text{ is } A_n^{\,r} \text{ AND } y \text{ is } B^r \text{ then } z = C^r \text{ for } r = 1, 2, 3, \ldots, M \quad (13.2)$$

where the vector x is input to the controller with components x_i, $i = 1, 2 \ldots n$, y and z are linguistic variables representing the process state variables and the controller output variable,

respectively, and A_i^r, B^r and C^r are the linguistic values of the linguistic variables x_i, y, and z in the universes of discourse, $U_1 \times U_2 \ldots \times U_n$, V, and W, respectively. The Equation (13.2) can be written compactly as

$$S^r : \text{if } x \text{ is } A^r \quad \text{AND} \quad Y \text{ is } B^r \text{ then } z = C^r \quad \text{for} \quad r = 1, 2, 3, \ldots, M \tag{13.3}$$

where A^r is the fuzzy cartesian product of the A_i^r's given by

$$A^r = A_1^r \times A_2^r \times A_3^r \times A_4^r \ldots A_n^r$$

The conditional rule for Sugeno Type controller can, similarly, be written in compact form as:

$$S^r : \text{if } x \text{ is } A^r \quad \text{AND} \quad y \text{ is } B^r \text{ then } z = f^r(x_1, x_2, \ldots x_n, y) \text{ for } r = 1, 2, 3, \ldots, M \tag{13.4}$$

where $f(.)$ is a function of the process state variables $(x_1, x_2, \ldots, x_n, y)$. The fuzzy control rules in Equation (13.3) and (13.4) evaluate the process state (*i.e.*, state, state error, state error integral, and so on) at time t and calculates and infers about the control actions as a function of the state variables $(x_1, x_2, \ldots, x_n, y)$. It is to be noted that both fuzzy control rules have linguistic values as inputs and either linguistic values [as in Equation (13.3)] or crisp values [as in Equation (13.4)] as outputs.

13.1.3 Contents of Antecedent and Consequent of Rules

The meaning of the contents of antecedent and consequent will be explained with reference to PID like fuzzy controller.

If we define the error as $e(k) = y_{sp} - y$, a linguistic value of *positive small* for the error means that the process output is below the set point but near to it so that the magnitude $|\ y_{sp} - y(k)\ |$ is small. Similarly linguistic value of *negative big* for the linguistic variable error means that the output is above the set point and the magnitude of $|\ y_{sp} - y(k)\ |$ is such that it is far away from the set point. A linguistic value of zero means that the process output is at the set point. Since, $\Delta e(k) = e(k) - e(k-1) = y(k-1) - y(k)$, negative $\Delta e(k)$ means that the process output $y(k)$ has increased when compared with its immediate past value $y(k-1)$.

Now suppose that the term sets (number of membership functions to cover the respective normalized domains) for $e(k)$, $\Delta e(k)$ and $u(k)$ are chosen the same and are given by {NB, NS, ZE, PS, PB} which are abbreviations for *Negative big, Negative small, Zero, Positive small* and *Positive big* respectively.

For a PD like fuzzy controller, the rule base is presented in a tabular form known as Fuzzy Associative Memory (FAM) table (see Table 13.1). The table entry corresponding to the first row and first column is PB and this represents a rule like

if $e(k)$ is NB and $\Delta e(k)$ is NB *then* $u(k)$ is PB

In order to understand the motivation behind such a rule, we note that the *NB error* means that the process state y is above the set point y_{sp} by a large magnitude and a negative sign for $\Delta e(k) = y(k-1) - y(k)$ means the output is *moving away* from the set point, so the *controller action* should be in the *opposite direction* with *large magnitude* to bring the output *towards* the set point. Therefore, it needs proper understanding of the process dynamics (expert knowledge) to write down the production rules. The entries in each cell of Table 13.1 is made based on this understanding.

13.1.4 Derivation of Production Rules

There is more than one approach for writing the production rules for a fuzzy controller, namely :

(*a*) Using expert experience and control engineering knowledge

(*b*) Modeling an operator's control actions

(*c*) Rules based on fuzzy model and behavior analysis of a controlled process

(*d*) Rules written using deterministic method

However, collecting expert knowledge is the most popular methods of writing production rules for fuzzy controller. This can be done either (*a*) by asking an expert to write down the production rules, in linguistic variables, for a given process, or (*b*) by extracting knowledge by a carefully prepared questionnaire from an experienced engineer or operator for subsequent verbalization of the rules.

This helps to build an initial prototype version of the production rules for a particular application, which needs tuning of the rule base and membership functions based on the closed loop response.

13.1.5 Membership Assignment

Database as part of knowledge base contains information about assignment of membership functions to input and output variables. This assignment can be intuitive or it can be based on some algorithmic or logical operation as discussed above.

The assignment of membership functions to inputs and outputs by *intuition*, among others, is the easiest to humans. It is simply derived from the capacity of humans to develop membership functions through their own innate intelligence and understanding of the physical world. Intuition involves contextual and semantic knowledge about an issue; it can also involve linguistic truth-values about this knowledge. We can intuitively assign linguistic values to the typical membership functions considered in Section 12.8 without much difficulty (vide Fig. 12.11 for example). The reader is referred to literature for other techniques of assigning membership functions.

13.1.6 Cardinality of a Term Set

The choice of term sets for linguistic variables was discussed in Section 12.8.1. Now the cardinality of the term set is a critical issue in realizing a fuzzy logic controller. By expanding the relations (13.3) and (13.4), it is apparent that even for a 3 input and one output system with cardinality of 7 for each of the three inputs, the total number of rules M will be 7 × 7 × 7 = 343, which is quite large and not easy to handle. So, some granularity may have to be sacrificed for keeping the value of M to a manageable size.

13.1.7 Completeness of Rules

Definition 13.1 A set of **if-then** rules is complete if any combination of input rules results in an appropriate output value. This means that

$$\forall x : \text{height}\ (u(x, y)) > 0 \tag{13.5}$$

where height of a fuzzy set is defined as $\text{hgt}\ (C) = \max_{u \in W} \mu_C(u)$

For a fuzzy logic controller with two inputs e and $\dot{e}$, one output u, each input having 5 term sets = {NB, NS, ZE, PS, PB}, the total number of rules will be equal to 5 × 5 = 25. A representative rule-set for PD like controller is shown in Table 13.1 below. A rule like "if e is NS and $\dot{e}$ is ZE then u is PS" has an entry PS in the table cell corresponding to the row NS in e and column ZE in $\dot{e}$. If any of the cell entries is void , *i.e.*, number of rules is less than 25, then the rule base is incomplete. The rule base for practical fuzzy logic controller is never complete. But it is not a problem in practice, since, some input combination is not realistic and certain domain of input is not of practical interest. For example, in a metro rail control system, simultaneous application of brake pressure and acceleration pressure is unrealistic. Fuzzy logic controller for inverted pendulum performs optimally with only 13 rules out of 25 rules.

13.1.8 Consistency of Rules

Definition 13.2. A set of **if-then** rules is **inconsistent** if there are rules r_1 and r_2 with the same rule antecedent and mutually exclusive rule-consequent.

As an illustration consider the following two rules from a collection of rule sets:

Rule 1 : IF e is PS and $\dot{e}$ is **NS then U is ZE**

Rule 2 : IF e is PS and $\dot{e}$ is **NS then U is NS**

In case of rule inconsistency, one of the rules is to be eliminated from the rule base.

Table 13.1 Representation of rule consequent for output u

Controller output		Rate of change of error, ė				
		NB	**NS**	**ZE**	**PS**	**PB**
error e	**NB**	PB	PB	PB	PS	ZE
	NS	PB	PB	PS	ZE	NS
	ZE	PB	PS	ZE	NS	NB
	PS	PS	ZE	NS	NB	NB
	PB	ZE	NS	NB	NB	NB

13.2 INFERENCE ENGINE

The task of the inference engine is to combine the fuzzy *if-then* rules for mapping the set A′ from the controller input space U to a fuzzy set B′ in the controller output space V using the production rules and the knowledge base of membership functions. In case of a single rule, the generalized modus ponen can be used for mapping A′ in U to B′ in V. However, in a real world control system numerous rules are needed to describe the control action and inferences can be drawn from the set of rules in two alternative ways: *composition based inference* and *individual rule based inference*.

(A) Composition based inference

In the composition based inference, all the fuzzy rules in the rule base are combined in a single fuzzy relation R in U × V which is then treated as a single *if-then* rule for the mapping of A′ in U to B′ in V. If each rule is considered as *independent conditional statements*, then the fuzzy relation computed from each rule by any implication functions including those given by relations (12.61)-(12.62), may be combined by *union* operation to get a single fuzzy relation. The single fuzzy relation, in turn is used to map A′ in U to B′ in V. If, on the other hand, the rules are thought of *strongly coupled conditional statements* such that all the rules must be satisfied simultaneously for proper control action, the final fuzzy relation may be obtained by the operation of *fuzzy intersection* of the individual rule implication relations. The resulting single fuzzy relation may then be used for mapping the fuzzy set A′ in U to set B′ in V.

(B) Individual rule based inference

In the individual rule based inference, the fuzzy set A′ in U is mapped to B′ in V by using the implication relation from each rule. Then the output of the whole fuzzy inference engine is found by combining, as before, the individual fuzzy sets either by *union or intersection* operation depending on the perception of coupling of conditional statements.

Though both types of inferences are conceptually feasible, we shall concentrate on the composition based inference engines in view of its popularity in the community of fuzzy control engineers. So, the fuzzy set at the controller output will be computed by using a value for R given in Equations (12.81) and (12.84).

Therefore, the design issues involved are the choice of :

(*i*) R from relation (12.81) and (12.84) for a given control system

(*ii*) popular implication functions from relations (12.61)-(12.62)

(*iii*) inference engine

(*iv*) and checking consistency and completeness of rules

In Section 12.14.1, we presented approximate reasoning procedure for computing the fuzzy set B′ [vide Equation (12.80)] as the conclusion from the fuzzy conditional rules given in Equation (12.78). We can adapt the same procedure for the calculation of the fuzzy set C′ as the conclusion for the control rules given in Equation (13.2) or (13.3). Let us consider the general form of MISO fuzzy control rules in the case of two input-single-output fuzzy systems as shown below.

$$
\begin{array}{lll}
 & S^1 & : \text{ If } x \text{ is } A^1 \text{ AND } y \text{ is } B^1 \text{ then } z \text{ is } C^1 \\
\text{also} & S^2 & : \text{ If } x \text{ is } A^2 \text{ AND } y \text{ is } B^2 \text{ then } z \text{ is } C^2 \\
 & & \quad \ldots \quad \ldots \quad \ldots \\
 & & \quad \ldots \quad \ldots \quad \ldots \\
\text{also} & S^m & : \text{ If } x \text{ is } A^m \text{ AND } y \text{ is } B^m \text{ then } z \text{ is } C^m \\
\text{Input} & & : x \text{ is } A' \text{ AND } y \text{ is } B' \\
\hline
\text{Conclusion} & & : z \text{ is } C'
\end{array}
\tag{13.6}
$$

In the above rules, the connectives AND and *also* may be interpreted as either intersection (∩) or union (∪) for different definitions of fuzzy implication. For the max-min compositional operator, *o* and the max product compositional operator, one can prove the following theorem [76].

Theorem 13.1. Considering only one rule R^1 in Equation (13.6), with Mamdani's min fuzzy implication, the conclusion C′, can be expressed as the intersection of the individual conclusions of input linguistic variables. That is,

$$
\begin{aligned}
C' &= (A', B') \; o \; R_M(A^r, B^r, C^r) \\
&= [A' \; o \; R_M(A^r, C^r) \cap [B \; o \; R_M(B^r, C^r)]
\end{aligned}
\tag{13.7}
$$

where $R_M(A^r, B^r, C^r) \equiv (A^r \text{ AND } B^r) \to C^r$, $R_M(A^r, C^r) = A^r \to C^r$ and $R_M(B^r, C^r) = B^r \to C^r$ are Mamdani fuzzy implications (12.61).

The next theorem, which is stated without proof, concerns the whole set of rules in Equation (13.6) and the interpretation of the connective *also* in the control rules in obtaining the conclusion.

Theorem 13.2. With Mamdani's min fuzzy implication R_M the conclusion C′ can be obtained for the whole set of rules in Equation (13.6) as a unification of the individual conclusions of fuzzy control rules. That is,

$$C' = (A', B')\ o \bigcup_{r=1}^{m} R_M(A^r, B^r, C^r) = \bigcup_{r=1}^{m} (A', B')\ o\ R_M(A^r, B^r, C^r)$$

$$= \bigcup_{r=1}^{m} \{[A'\ o\ R_M(A^r, C^r)] \cap [B'\ o\ R_M(B^r, C^r)]\} \quad (13.8)$$

13.2.1 Special Cases of Fuzzy Singleton

When the inputs considered are fuzzy singletons, which are usually the case in fuzzy control systems, we can get the conclusion set C′ in a more simplified form by using two special relations—the Mamdani fuzzy implication relation R_M and the Larsen (or product) implication relation R_P. For singleton fuzzifier [vide Section 13.4.1], $f(x_0) = 1$ and $A' = u_0$ for $x = x_0$ and $B' = v_0$ for $y = y_0$, consequently, μ_A^r and μ_B^r will be scalar quantities α_A^r, α_B^r respectively, so the antecedent of the r-th rule in (13.6) reduces to $\alpha_A^r \wedge \alpha_B^r = \alpha^r$, where α^r is known as firing strength of the r-th rule [vide Fig. 13.2]. Using these values and max-min compositional operator o for all the rules in Equation (13.6), we get the results stated in Theorem 13.3 below, which plays a very important role in fuzzy logic controller implementation.

Theorem 13.3. If the inputs are fuzzy singletons, namely, $A' = u_0$ and $B' = v_0$, then the results C′ in Equation (13.6) derived by employing Mamdani's minimum operation rule R_M and Larsen's product operation rule R_P, respectively, may be expressed as

$$R_M : \mu_{C'}(w) = \bigvee_{r=1}^{m} \alpha^r \wedge \mu_C{}^r(w) = \bigvee_{r=1}^{m} [\mu_A{}^r(u_0) \wedge \mu_B{}^r(v_0)] \wedge \mu_C{}^r(w) \quad (13.9)$$

$$R_P : \mu_{C'}(w) = \bigvee_{r=1}^{m} \alpha^r . \mu_C{}^r(w) = \bigvee_{r=1}^{m} [\mu_A{}^r(u_0) \wedge \mu_B{}^r(v_0)] . \mu_C{}^r(w) \quad (13.10)$$

where α^r denotes the firing strength of the r-th rule, which is a measure of the contribution of the r-th rule to the fuzzy control action.

Equations (13.9) and (13.10) are the most frequently used in fuzzy control applications. They not only simplify computations but also provide a graphical interpretation of the fuzzy inference mechanism in the FLC.

In order to get a better feel, we write the relations (13.9) and (13.10) in expanded form for the case with only $r = 2$ rules, which appear as :

$$\mu_{C'}(w) = \bigvee_{r=1}^{2} \alpha^r \wedge \mu_C{}^r(w) = [\alpha^1 \wedge \mu_C{}^1(w)] \vee [\alpha^2 \wedge \mu_C{}^2(w)]\ ; \quad (13.11)$$

where $\alpha^1 = \mu_A{}^1(u_0) \wedge \mu_B{}^1(v_0)$ and $\alpha^2 = \mu_A{}^2(u_0) \wedge \mu_B{}^2(v_0)$

$$\mu_{C'}(w) = \bigvee_{r=1}^{2} \alpha^r . \mu_C{}^r(w) = [\alpha^1 . \mu^1{}_C(w)] \vee [\alpha^2 . \mu^2{}_C(w)]\ ; \quad (13.12)$$

where $\alpha^1 = \mu_A{}^1(u_0) . \mu^1{}_B(v_0)$ and $\alpha^2 = \mu^2{}_A(u_0) . \mu_B{}^2(v_0)$

In the above relations α^1 and α^2 are the firing strengths of the rules 1 and 2 in relation (13.6) respectively. We can extend the fuzzy approximate reasoning outlined in Section 12.13 and illustrate the relations (13.11) and (13.12) graphically as shown in Fig. 13.2(a) and 13.2(b) respectively.

13.3 REASONING TYPES

For future reference we shall consider the following types of fuzzy reasoning for implementation of fuzzy logic controllers

(A) **Fuzzy reasoning of type 1** with Mamdani implication given by relation (13.11)

The process is illustrated in Fig. 13.2(*a*)

Fuzzy reasoning of type 1 with Larsen implication given by relation (13.12)

The process is illustrated in Fig. 13.2(*b*)

(B) **Fuzzy reasoning of type 2 (Takagi-Sugeno type[48-49])** : *The consequent of the rule is a function of the system state variables.* In this mode of reasoning the fuzzy conditional rules are of the form given by (13.4). Considering only 2 rules for the purpose of illustration, Equation (13.4) may be written as :

S^1 : If x is A^1 and y is B^1 then z is $f^1(x, y)$

S^2 : If x is A^2 and y is B^2 then z is $f^2(x, y)$

The inferred values from rule 1 and rule 2 are $\alpha^1 f^1(x_0 y_0)$ and $\alpha^2 f^2(x_0 y_0)$ respectively. Consequently a crisp output value may be obtained as

$$z_0 = \frac{\alpha^1 f^1(x_0, y_0) + \alpha^2 f^2(x_0, y_0)}{\alpha^1 + \alpha^2} \tag{13.13}$$

For m number of rules the crisp output is given by $z_0 = \dfrac{\sum_{r=1}^{m} \alpha^r f^r(x_0, y_0)}{\sum_{r=1}^{m} \alpha^r}$ (13.14)

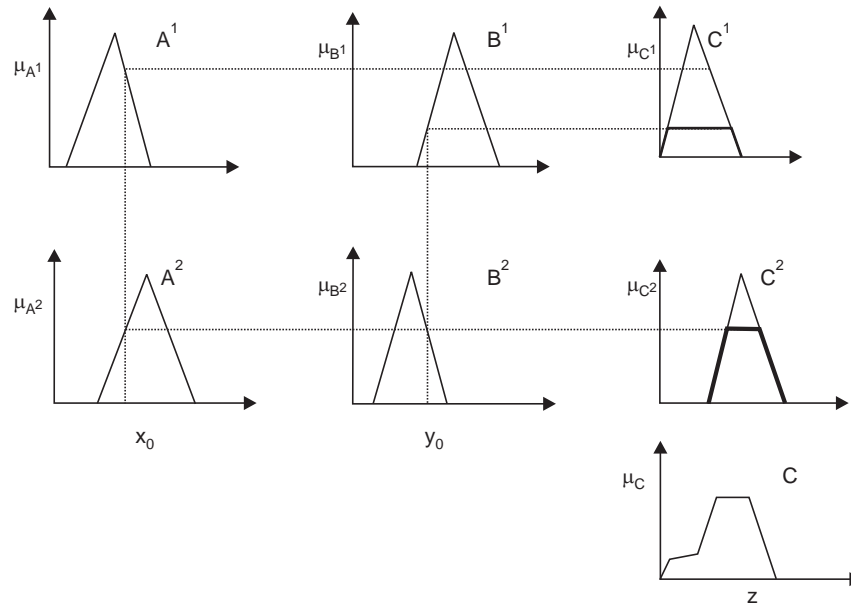



Fig. 13.2 (*a*) Representation of fuzzy reasoning of the type 1 with Mamdani implication

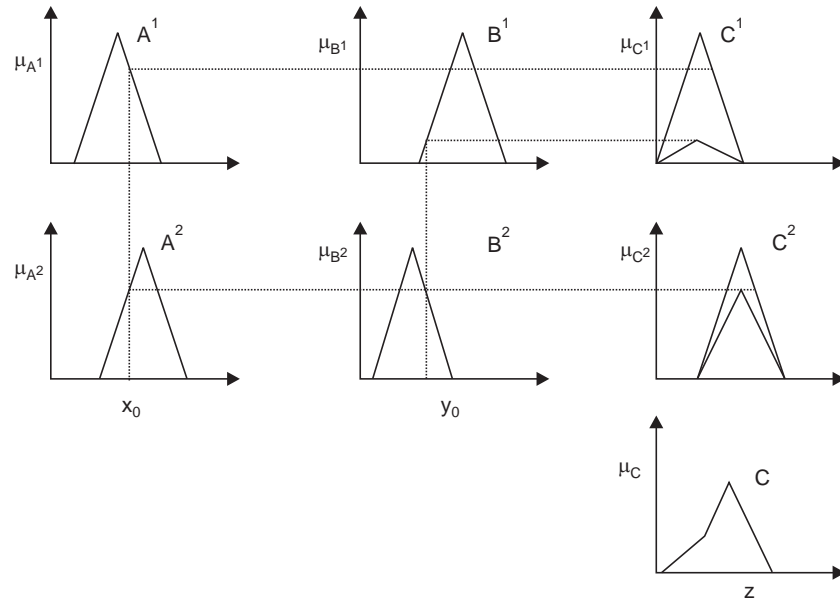



Fig. 13.2 (*b*) Representation of fuzzy reasoning of the type 1 with Larsen product implication

13.4 FUZZIFICATION MODULE

The Fuzzification Module performs the following basic tasks :

It performs input normalization through scale transformation mapping the physical values of the state variables into a normalized universe of discourse. It also maps the normalized value of the controller output variable onto physical domain.

It converts a point wise crisp value of the process state variable to make it compatible with the fuzzy set representation of the system state variable in the rule-antecedent.

Fuzzification is done based on the type of the inference engine or the strategy of inference like disjunction rule-based or composition based.

13.4.1 Fuzzifier and Fuzzy Singleton

Fuzzification is the process of converting a crisp input to a fuzzy value. This is done because many quantities that we consider crisp and deterministic, contain considerable uncertainty and can be better represented by fuzzy linguistic values. Since the data manipulation in a FLC is based on fuzzy set theory, fuzzification is necessary and desirable at an early stage. Hence, *fuzzifier can be defined as a mapping from an observed input space to labels of fuzzy sets in a specified input universe of discourse*.

The mapping function takes care of the associated measurement uncertainty for each input variable. The purpose of the mapping function is to interpret measurements of input variables, each expressed by a real number as more realistic fuzzy approximations of the respective real numbers. If f is the mapping function applied to a variable x it can be written as :

$$f(x_i) : [-k, +k] \rightarrow R^+$$

where R^+ is the set of fuzzy numbers and $f(x_0)$ is a real number chosen by f as a fuzzy approximation of the measurement $x_i = x_0$. A possible definition for this fuzzy number for any $x_i \in [-a, +a]$ is shown in Fig. 13.3(a), where ε is a parameter to be determined in the context

of each application. If it is desirable, other shapes of mapping functions may be used for the fuzzy number $f(x_0)$. For each measurement $x_i = x_0$ the fuzzy set $f(x_0)$ enters into the inference mechanism.

A natural and simple fuzzification approach, however, is to convert a crisp value x_0 into a *fuzzy singleton* A within the specified universe of discourse, since in fuzzy control applications, the observed data are usually crisp. The membership function of fuzzy singleton A, $\mu_A(x_0)$, is equal to 1 at the point x_0 and zero elsewhere (see Fig. 13.3(*b*)). When a singleton fuzzifier is used for a measured state $x_i(t)$ at time t, it is mapped to the linguistic term set $T^1_{x_i}$ with degree $\mu^1_{x_i}(x_i(t))$ and to the term set $T^2_{x_i}$ with degree $\mu^2_{x_i}(x_i(t))$ and so on. This approach is widely used in FLC applications because it greatly simplifies the fuzzy reasoning process.

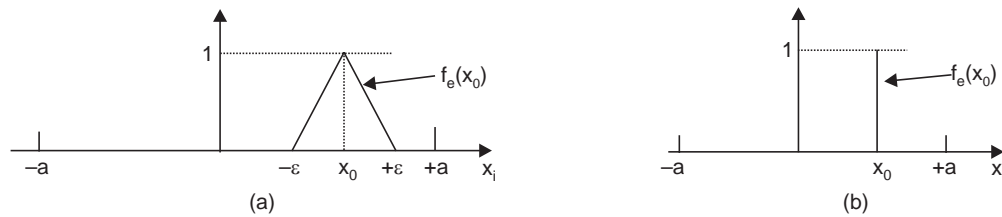



Fig. 13.3 Fuzzifier functions (*a*) a triangular fuzzification function, (*b*) singleton fuzzification function.

13.5 DEFUZZIFICATION MODULE

The tasks of defuzzification module are to:

(*a*) convert the set of controller output values into a single point wise value

(*b*) perform output renormalization that maps the point wise value of the controller output into its physical domain.

The choice of defuzzification operator [50] is the most important consideration in the defuzzification stage.

13.5.1 Defuzzifier

Conceptually, the task of the defuzzifier is to specify a point in W that best represents the fuzzy set C′, obtained as the output from the inference engine. So, defuzzification may be defined as a mapping of the set C′ from a space of fuzzy control actions in the output universe of discourse [51], $W \subset R$ into a space of crisp control actions $z^* \in W$. This process is necessary because in many practical applications crisp control action is required to actuate the control and it is essential in the reasoning of the first type [Equations (13.6) and (13.7)] .

There are a number of choices in determining this crisp controller output z^*, however, the following three criteria should be considered in choosing a defuzzification scheme:

1. Plausibility: The point z^* should represent C′ from an intuitive point of view; for example, it may lie approximately in the middle of the support of C′ or has a high degree of membership in C′.

2. Computational simplicity: This criterion is particularly important for fuzzy control because fuzzy controllers operate in real-time.

3. Continuity: A small change in C′ should not result in a large change in z^*.

4. Disambiguity: It means that the defuzzification method should always produce an unique value for z^*.

For all the defuzzifiers, it is assumed that the fuzzy set C′ is obtained from a **fuzzy reasoning of type 1** that employs one of the implication functions given in the Equations (12.61)-(12.62) such that C′ is the union or intersection of M individual fuzzy sets. There is no need of a defuzzifier for the **fuzzy reasoning of type 2 or Tagaki-Sugeno type** where the crisp output is found from equation of the form given in Equation (13.14)

We now present some widely used types of defuzzifiers .

13.5.2 Center of Area (or Center of Gravity) Defuzzifier

The center *of gravity* defuzzifier specifies the z^* as the center of the area (COA) covered by the membership function of C′ and is given by equation of the form.

$$z^* = \frac{\int_w \mu_C(z)\, z dz}{\int_w \mu_C(z)\, dz} \tag{13.15}$$

where $\int_W$ is the conventional integral. Fig. 13.4 shows this operation graphically.

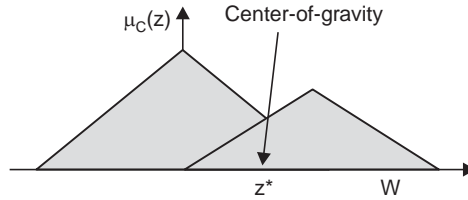



Fig. 13.4 A graphical representation of the center of gravity defuzzifier

For the discrete case in which $\mu_c(z)$ is defined on a universal set C = $\{z_1, z_2, \ldots z_n\}$, the COA method (13.15) yields

$$z^* = \frac{\sum_{j=1}^{n} \mu_c(z_j)\, z_j}{\sum_{j=1}^{n} \mu_c(z_j)} \tag{13.16}$$

where n is the number of quantization levels of the output, z_j is the amount of control output at the quantization level j, and $\mu_c(z_j)$ represents its membership value in the output fuzzy set C. If z^* is not equal to any value in the universal set C, we take the value closest to it. We note that

$$\frac{\mu_c(z_j)}{\sum_{j=1}^{n} \mu_c(z_j)}$$

for all j = 1, 2, ... n, form a probability distribution obtained from the membership functions $\mu_c(z_j)$ by the ratio scale transformation. Therefore, the defuzzified value z^* obtained from Equation (13.16) can be interpreted as the expected value of z.

The advantage of the center of gravity defuzzifier lies in its intuitive plausibility. The disadvantage is that it is computationally intensive. In fact, the membership function $\mu_c'(z)$, is usually irregular and therefore the integrations in Equation (13.15) is difficult to compute. The next defuzzifier tries to overcome this disadvantage by approximating Equation (13.15) with a simpler formula.

13.5.3 Center Average Defuzzifier (or Weighted Average Method)

This method is only *valid for symmetrical output membership functions*. Since, the fuzzy set C′ is the union or intersection of M fuzzy sets, a good approximation of (13.15) is the weighted average (mean) of the centers of the M fuzzy sets, with the weights equal the heights (maximum) of the corresponding fuzzy sets. Specifically, let $\bar{z}_r$ be the center of the r-th fuzzy set and w_r be its height, the center average defuzzifier determines z^* as

$$z^* = \frac{\sum_{r=1}^{m} \bar{z}_r\, w_r}{\sum_{r=1}^{m} w_r} \tag{13.17}$$

Figure 13.5 illustrates this operation graphically for a simple example with M = 2, where $\bar{z}_1 = 3$ and $\bar{z}_2 = 5$.

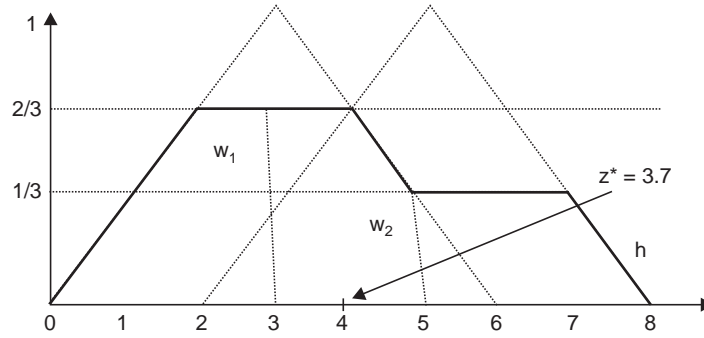



Fig. 13.5 A graphical representation of the center average defuzzifier

The center average defuzzifier is the most commonly used defuzzifier in fuzzy systems and fuzzy control. It is computationally simple and intuitively plausible. Also, small changes in $\bar{z}_r$ and w_r result in small changes in z^*. We now illustrate the center of gravity and center average defuzzifier for a simple example.

Example 13.1 Suppose that the fuzzy set C′ is represented by the solid curve in Fig. 13.5 and is described by the function in Equation (13.18).

$$\begin{aligned} \mu_C(z) &= \frac{z}{3}, \quad 0 < z \le 2 \\ &= \frac{2}{3}, \quad 2 < z \le 4 \\ &= \frac{6-z}{3}, \quad 4 < z \le 5 \\ &= \frac{1}{3}, \quad 5 < z \le 7 \\ &= \frac{8-z}{3}, \quad 7 < z \le 8 \end{aligned} \tag{13.18}$$

For computing the defuzzified value by the COA method, we enumerate the numerator of the Equation (13.15) first as :

$$\text{Num} = \int_0^2 \frac{z}{3} z\,dz + \int_2^4 \frac{2}{3} z\,dz + \int_4^5 \frac{6-z}{3} z\,dz + \int_5^7 \frac{1}{3} z\,dz + \int_7^8 \frac{8-z}{3} z\,dz$$

$$= \frac{1}{3}\frac{z^3}{3}\Bigg|_0^2 + \frac{2}{3}\frac{z^2}{2}\Bigg|_2^4 + 2\frac{z^2}{2}\Bigg|_4^5 - \frac{1}{3}\frac{z^3}{3}\Bigg|_4^5 + \frac{1}{3}\frac{z^2}{2}\Bigg|_5^7 + \frac{8}{3}\frac{z^2}{2}\Bigg|_7^8 - \frac{1}{3}\frac{z^3}{3}\Bigg|_7^8 = \frac{37}{3}$$

Similarly the denominator of Equation (13.15) is computed as

$$\text{Deno} = \int_0^2 \frac{z}{3} dz + \int_2^4 \frac{2}{3} dz + \int_4^5 \frac{6-z}{3} dz + \int_5^7 \frac{1}{3} dz + \int_7^8 \frac{8-z}{3} dz$$

$$= \frac{1}{3}\frac{z^2}{2}\Bigg|_0^2 + \frac{2}{3}z\Bigg|_2^4 + 2z\Bigg|_4^5 - \frac{1}{3}\frac{z^2}{2}\Bigg|_4^5 + \frac{1}{3}z\Bigg|_5^7 + \frac{8}{3}z\Bigg|_7^8 - \frac{1}{3}\frac{z^2}{2}\Bigg|_7^8$$

$$= \frac{10}{3}$$

Hence $\quad z^* = \text{Num/Deno} = 3.7$ (13.19)

Let us now find the defuzzified value by applying the Equation (13.16) by considering the discrete levels of z ={1, 2, 3, 4, 5, 6, 7} for which the corresponding discretized value of $\mu_c(z) = \left\{\frac{1}{3}, \frac{2}{3}, \frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right\}$ are noted from the graph in Fig. 13.5. Application of the Equation (13.16) yields

$$z^* = \frac{1\left(\frac{1}{3}\right) + 2\left(\frac{2}{3}\right) + 3\left(\frac{2}{3}\right) + 4\left(\frac{2}{3}\right) + 5\left(\frac{1}{3}\right) + 6\left(\frac{1}{3}\right) + 7\left(\frac{1}{3}\right)}{\left(\frac{1}{3}\right) + \left(\frac{2}{3}\right) + \left(\frac{2}{3}\right) + \left(\frac{2}{3}\right) + \left(\frac{1}{3}\right) + \left(\frac{1}{3}\right) + \left(\frac{1}{3}\right)} = \frac{\frac{37}{3}}{\frac{10}{3}} = 3.7$$

which happens to be the same found by treating the membership function for z as a continuous function. We note from the Fig. 13.5 that $w_1 = 2/3$, $w_2 = 1/3$, $\bar{z}_1 = 3$ and $\bar{z}_2 = 5$. Therefore, applying the Equation for (13.17) for the center of average defuzzifier, we get

$$z^* = \frac{3\left(\frac{2}{3}\right) + 5\left(\frac{1}{3}\right)}{\frac{2}{3} + \frac{1}{3}} = \frac{\frac{11}{3}}{1} = 3.667$$

Of the above two commonly used defuzzification strategies, the COA strategy has been reported to yield superior steady state performance of the closed loop system.

3.6 DESIGN CONSIDERATION OF SIMPLE FUZZY CONTROLLERS

Simple fuzzy controllers can generally be depicted by a block diagram shown in Fig. 13.6. We can write down the steps involved in designing a simple fuzzy control system as follows :

(*a*) Identify the inputs, states, and outputs variables of the plant.

(*b*) Partition the interval spanned by each variable into a number of fuzzy partitions or term sets, assigning each a linguistic label.

(*c*) Assign a membership function to each fuzzy term set.

(*d*) Form the rule-base by assigning the fuzzy relationships between the input fuzzy term sets and the output fuzzy term sets.

(*e*) Normalize the input and output variables to the interval [0, 1] or the [– 1, 1] by proper choice of scaling factors for them.

(*f*) Fuzzify the inputs to the controller.

(*g*) Infer the controller output contributed from each rule by implication relation.

(*h*) Combine the fuzzy outputs recommended by each rule.

(*i*) Find crisp controller output by de-fuzzification.

If the control law is Sugeno-Tagaki type, no defuzzification is necessary.

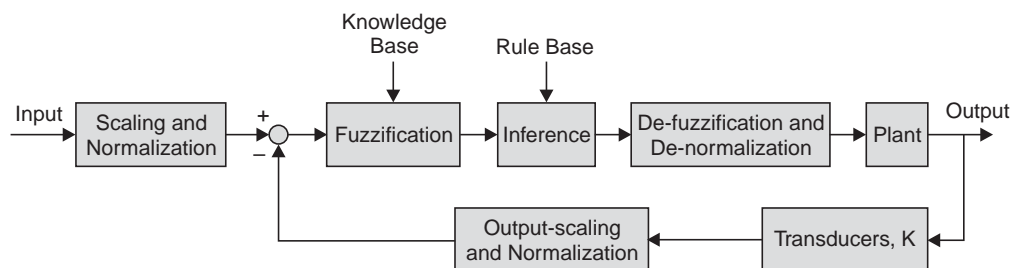



Fig. 13.6 Block diagram of a simple fuzzy control system

If the simple fuzzy logic controller is non-adaptive, the control law derived by the steps mentioned above is fixed. However, in an adaptive fuzzy logic controller, the control laws are adaptively modified based on some adaptation law in order to improve system performance.

A *simple fuzzy* control system is characterized with the following features : It has :

(*a*) Fixed and uniform input- and output-scaling factors.

(*b*) The rule-base is fixed and all the rules are assigned the same degree of certainty and confidence, which is equal to unity.

(*c*) Fixed membership functions with fixed term sets.

(*d*) Limited number of rules, which increases exponentially with the number of input control variables.

(*e*) Fixed meta-knowledge including the implication relation, rules aggregation, and output de-fuzzification.

(*f*) Low-level control without any hierarchical rule structure.

13.7 DESIGN PARAMETERS OF GENERAL FUZZY CONTROLLERS

The principal design strategy in a *general fuzzy controller* may involve the following steps [52-55] :

1. Fuzzification strategies
2. Knowledge base :

   (*a*) Normalization of the input and output spaces

(*b*) Choice of term sets in the input and output spaces

(*c*) Choice of the membership functions of the input/output control variables.

3. Rule-base:

(*a*) Choice of control variables and process state variables

(*b*) Policy for the derivation of fuzzy control rules

(*c*) Types of fuzzy controller

(*d*) Consistency and completeness and adaptation of fuzzy control rules

4. Decision-making logic:

(*a*) Choice of a fuzzy implication

(*b*) Interpretation of the sentence connective **'and'** and **'or'**

(*c*) Inference Type

5. The choice of a defuzzifier

For a simple fuzzy controller, all the above five steps are fixed, whereas in an adaptive fuzzy control system one or more of the steps are adapted for improving the controller performance.

Most Fuzzy logic control system models can be expressed in two different forms: fuzzy rule-based structures and fuzzy relational equations. We have already discussed the most commonly used fuzzy rule-based structures in fuzzy control systems in the earlier sections. We shall now present some illustrative examples of designing fuzzy controllers.

13.8 EXAMPLES OF FUZZY CONTROL SYSTEM DESIGN : INVERTED PENDULUM

Figure 13.7 shows the classic inverted pendulum system, which has been an interesting case in control theory for many years.

Example 13.2 *Let us design a fuzzy controller for the simplified version of the inverted pendulum system shown in Fig. 13.7. The linearised differential equation describing the dynamics of the system is given by the equation* (*A.159*) *in Appendix A, which is reproduced below:*

$$\frac{4l}{3}\frac{(4M+m)}{4m}\ddot{\theta} - \frac{(M+m)g}{m}\theta = -\frac{u}{m} \tag{13.20}$$

where m = the mass of the pole assumed to be concentrated at the center of the pendulum, Kg

M = mass of the cart, Kg

$2l$ = the length of the pendulum, meter

θ = the deviation angle from vertical in the clockwise direction, in radian

$T(t)$ = the torque applied to the pole in the counterclockwise direction, Nm

$u(t)$ = the control on the cart acting from left to the right producing the counterclockwise torque $T(t)$, Nm

t = time in sec, and

g = the gravitational acceleration constant

A simplified model of the system dynamics, without the cart, is available in Reference [56].

When θ is expressed in degrees, the coefficient of u is to be multiplied by $180/\pi$. For convenience of hand calculation, we choose the following parameters for the system in equation (13.20) where θ is expressed in degrees and $\dot{\theta}$ in degrees per second.

$$M = \frac{180}{\pi g} - m \quad \text{and} \quad l = \frac{3\,(M + m)\,g}{4M + m} \tag{13.21}$$

With the above choices of the parameters, the equation (13.20) reduces to $\ddot{\theta} = \theta - u$ (13.22)

With $x_1 = \theta$ and $x_2 = \dot{\theta}$ as state variables, the state-space representation for the linearized system defined by Equation (13.22) is given by

$$dx_1/dt = x_2 \tag{13.23}$$

$$dx_2/dt = x_1 - u(t) \tag{13.24}$$

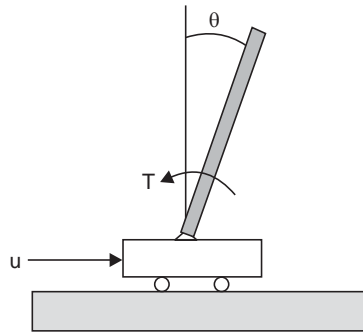



Fig. 13.7 Inverted pendulum control problem

The discrete-time state-space equations can be represented as matrix difference equations, with sampling time T set to 1 sec as :

$$x_1(k + 1) = x_1(k) + Tx_2(k) = x_1(k) + x_2(k) \tag{13.25}$$

$$x_2(k + 1) = Tx_1(k) + x_2(k) - Tu(k) = x_1(k) + x_2(k) - u(k) \tag{13.26}$$

For this problem we assume the universe of discourse for the two variables to be $-2° \le x_1 \le 2°$ and -4 dps $\le x_2 \le 4$ dps [dps = degree per second].

Step 1. We decide to use 3 term sets to cover the universe of discourse for both the states x_1 and x_2. The term sets are designated as positive (P), zero (Z), and negative (N). The membership functions of the three term set functions for x_1 on its universe are as shown in Fig. 13.8. Similarly the membership functions of the three term set functions for x_2 on its universe are shown in Fig. 13.9.

Step 2. In contrast to 3 term sets for the inputs to the fuzzy controller (the states x_1, x_2), we decide to use 5 term sets for the output of the controller, $u(k)$, designated as Negative Big (NB), Negative (N), Zero (Z), Positive (P), Positive Big (PB) respectively on its universe, which is $-10 \le u(k) \le 10$. The membership functions of the 5 term sets are shown in Fig. 13.10.

Step 3. We construct nine rules in a 3 × 3 Fuzzy Associated Memory (FAM) Table 13.2, for this system, which would involve θ and $\dot{\theta}$ for balancing the inverted pendulum in the vertical position. The Rule numbers are indicated in the parenthesis in the table cells along with the control actions, $u(k)$.

Chapter 13

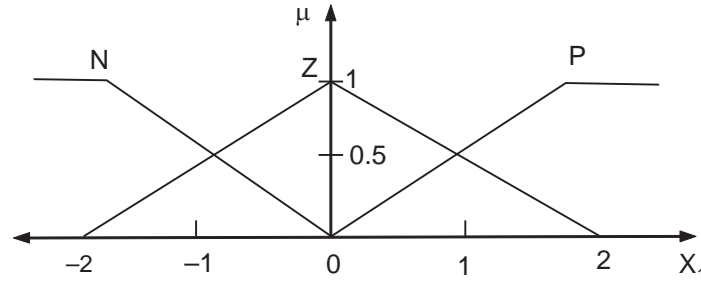



Fig. 13.8 Input x_1 partitioned

Step 4. We shall now prepare a simulation of the controller using the rules in the FAM Table 13.2 starting with initial conditions:

$$x_1(0) = 1° \text{ and } x_2(0) = 0 \text{ dps}$$

We shall employ graphical methods for computing controller outputs using the rules and the plant Equations (13.25) and (13.26). The final controller outputs are summarized for few discrete steps $0 \leq k \leq 12$ in Table 13.3.

In each discrete step of simulation, we will find the firing strengths of the rules, which yields the controller output by the rules in the FAM Table 13.2. The control action $u(k)$ is computed by using Mamdani implication. The membership function is defuzzified for the control action using the center of area or centroid method. The recursive difference equations is then solved for new values of x_1 and x_2 to be used as the initial conditions to the next discrete step of the recursive solution.

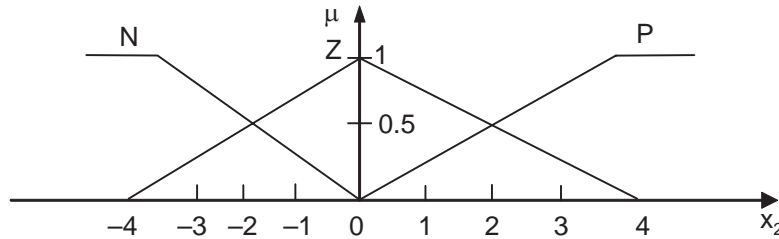



Fig. 13.9 Input x_2 partitioned

With reference to Figs. 13.8 and 13.9, we note that with singleton initial conditions for $x_1(0) = 1°$ and $x_2(0) = 0$ dps respectively, rules 5 and 6 in Table 13.2 are activated (see also Fig. 13.11).

Rule 5 : If x_1 is Z and x_2 is Z then u = Z, So, the firing strength,

$$\alpha^1 = \min (0.5, 1) = 0.5 \; ; \; u = 0.5(\text{Z})$$

Rule 6 : If x_1 is P and x_2 is Z then u = P, with firing strength,

$$\alpha^2 = \min (0.5, 1) = 0.5 \; ; \; u = 0.5(\text{P})$$

The firing of the rules are graphically depicted in Fig. 13.11. In part (*c*) of Fig. 13.11, the union of the clipped fuzzy consequents are shown for the control variable, u. The final form,

with the defuzzified control value, is shown at the bottom of Fig. 13.11(*c*). The defuzzified value of u may be computed by using Equation (13.17) after quantization of the domain u from – 10 to + 10 , which yields $u^* = 2.50$.

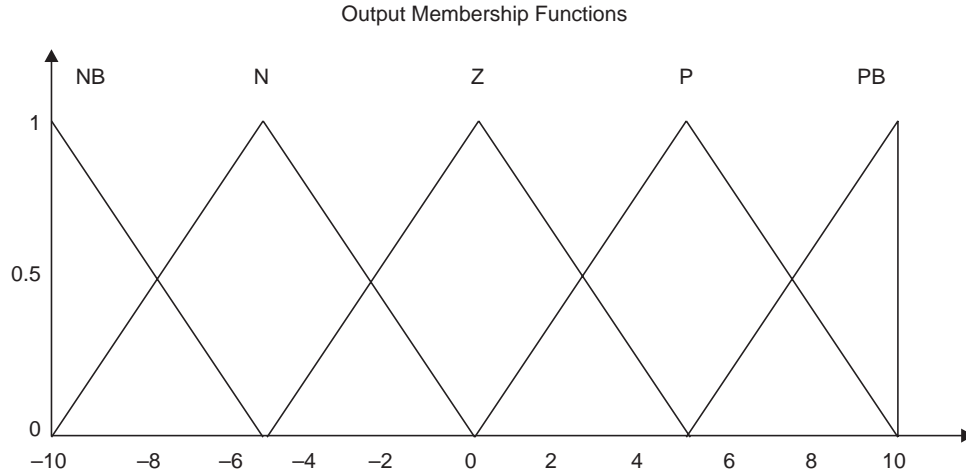



Fig. 13.10 Output variable u, partitioned

Table 13.2 FAM table

x_1 \ x_2	P	Z	N
P	PB (3)	P (6)	Z (9)
Z	P (2)	Z (5)	N (8)
N	Z (1)	N (4)	NB (7)

This completes the first discrete step of simulation. Now taking the value of the defuzzified control variable, $u^* = 2.50$ the initial conditions for next iteration is found, with the help of state equation (13.25) and (13.26) as:

$$x_1(1) = x_1(0) + x_2(0) = 1 + 0 = 1$$

$$x_2(1) = x_1(0) + x_2(0) - u(0) = 1 + 0 - 2.5 = -1.50$$

So, the initial conditions for the second discrete step are $x_1(1) = 1$ and $x_2(1) = -1.50$, respectively and are shown graphically in Fig. 13.12(*a*) and 13.12(*b*). From the Table 13.2, we note that for this initial conditions, Rule 5, Rule 6, Rule 8 and Rule 9 are fired. The Rules along with their firing strengths α's are shown below:

Rule 5 : If x_1 is Z and x_2 is Z then u is Z, $\alpha^1 = \min(0.5, 0.625) = 0.5$, $u = 0.5(Z)$

Rule 6 : If x_1 is P and x_2 is Z then u is P, $\alpha^2 = \min(0.5, 0.625) = 0.5$, $u = 0.5(P)$

Rule 8 : If x_1 is Z and x_2 is N then u is N, $\alpha^3 = \min(0.5, 0.375) = 0.375$, $u = 0.375(N)$

Rule 9 : If x_1 is P and x_2 is N then u is Z, $\alpha^4 = \min(0.5, 0.375) = 0.375$, $u = 0.375(Z)$

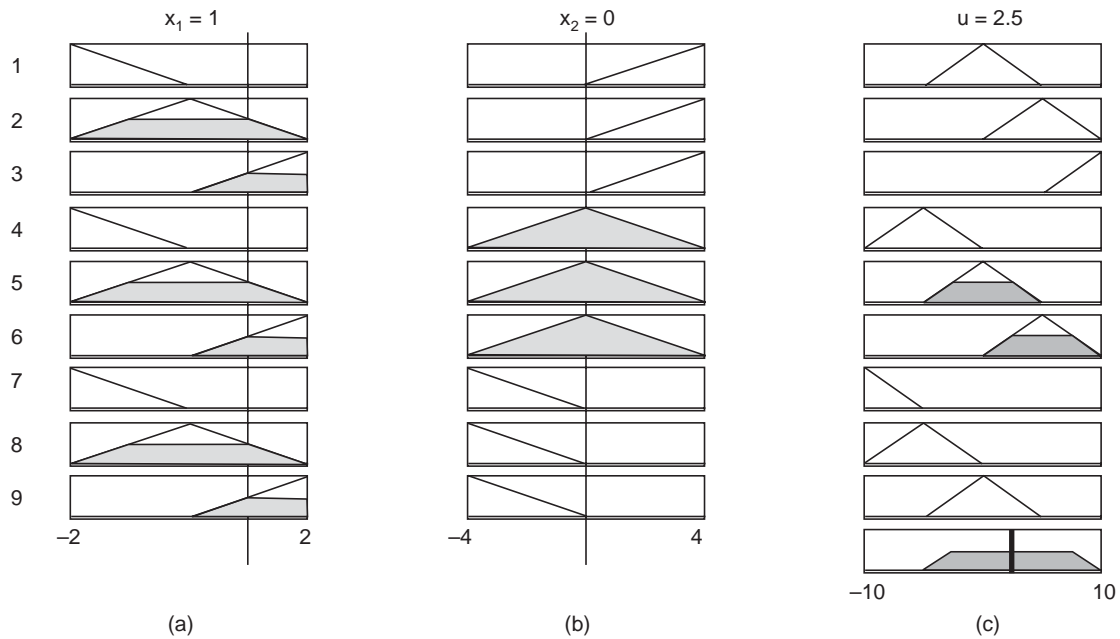



Fig. 13.11 Firing of rules with initial values of $x_1(0)=1$ and $x_2(0) = 0$ and the defuzzified output u = 2.5

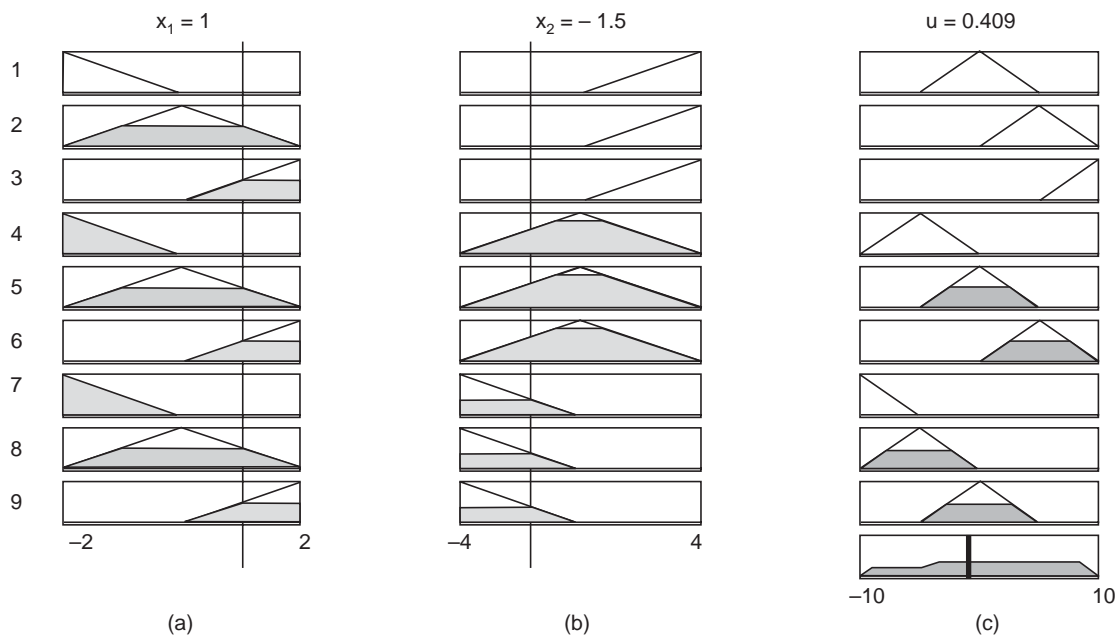



Fig. 13.12 Firing of rules with initial values of $x_1(1) =1$ and $x_2(1) = – 1.5$ and the defuzzified output u = 0.409

The union of the fuzzy consequents and the resulting defuzzified output are shown at the bottom of Fig. 13.12(*c*). The defuzzified value (using Equation 13.17) is $u^* = 0.409$.

We now use $u = 0.409$ to find the initial conditions for third discrete iterative step.

$$x_1(2) = x_1(1) + x_2(1) = 1 - 1.5 = -0.50$$

$$x_2(2) = x_1(1) + x_2(1) - u(t) = 1 - 1.5 - 0.409 = -0.909$$

Thus, with initial conditions $x_1(2) = -0.50$ and $x_2(2) = -0.909$, we can proceed as before using the Table 13.2 and graphically compute the defuzzified controller output as $u^* = -1.65$. The results up to 12 cycles are shown in tabular form in Table 13.3

Table 13.3

k	$x_1(k)$	$x_2(k)$	$u(k)$
0	1	0	2.500
1	1	– 1.500	0.409
2	– 0.500	– 0.909	– 1.650
3	– 1.409	0.241	– 2.840
4	– 1.168	1.672	– 0.501
5	0.504	1.005	1.700
6	1.509	– 0.191	3.150
7	1.318	– 1.832	0.670
8	– 0.514	– 1.184	– 1.900
9	– 1.698	0.202	– 3.560
10	– 1.496	2.064	– 0.875
11	0.568	1.443	2.210
12	2.011	– 0.199	4.650

Figure 13.13 shows the rule firing with initial conditions $x_1(3) = -1.409$ and $x_2(3) = 0.241$ to get a control action of $u^* = -2.840$ in the 3rd cycle. For the 12th cycle, the rule firing with initial conditions $x_1(12) = 2.011$ and $x_2(12) = -0.199$ is shown in Fig. 13.14 to get a control action of $u^* = 4.650$.

It is to be noted that, because of the chosen domain of x_1 to $-2 < x_1 < +2$, the initial condition 2.011 appears as + 2 in the Fig. 13.14(*a*).

Step 5. We now plot the four simulation cycle results for x_1, x_2, and $u(k)$. In Figures 13.15 to 13.17, the length and direction of the arrow are proportional to the angular velocity and show direction of motion of the pendulum, respectively.

We have endeavored to give an introductory exposition of this new discipline of fuzzy control which is influencing currently the industrial scenario in a major way and has found its way in most domestic appliances. The interested readers are referred to the literature listed at the end of this book to learn more about this emerging discipline.

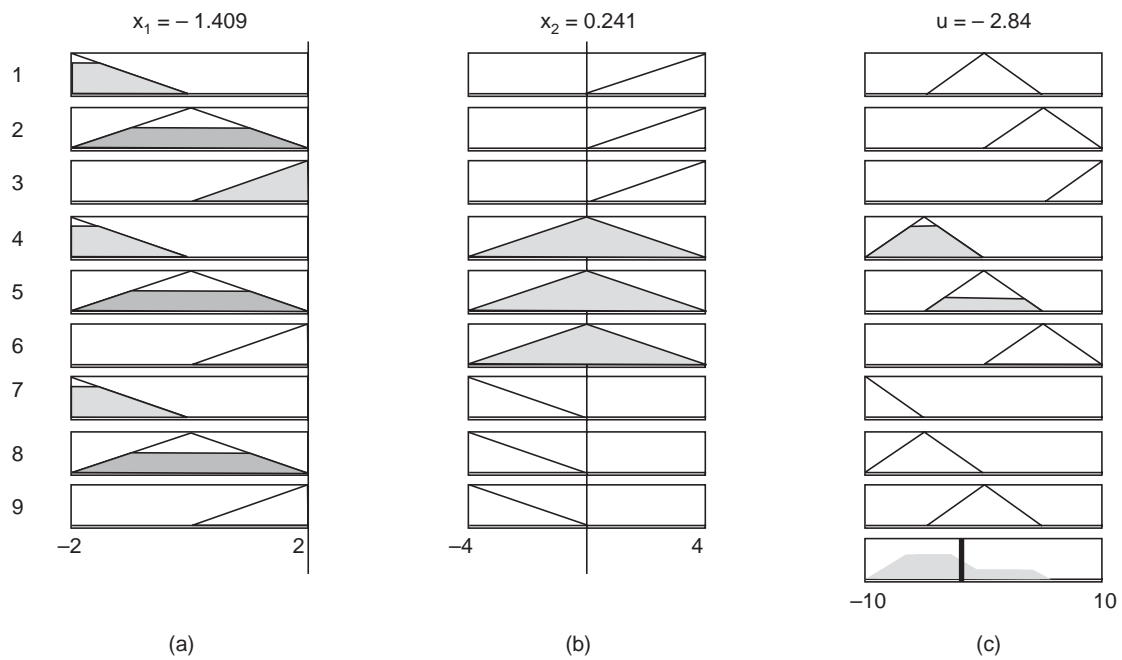



Fig. 13.13 Firing of rules with initial values of $x_1(3) = -1.409$ and $x_2(3) = 0.241$ and the defuzzified output $u^* = -2.84$.

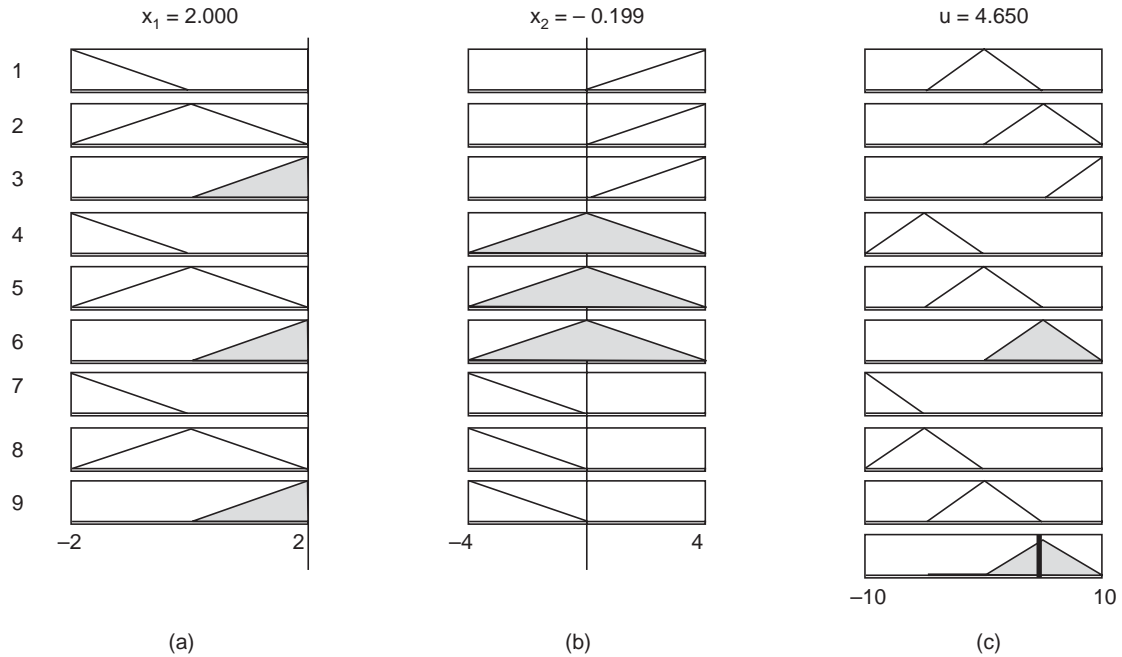



Fig. 13.14 Firing of rules with initial values of $x_1(12) = 2.011$ and $x_2(12) = -0.199$ and the defuzzified output $u^* = 4.650$

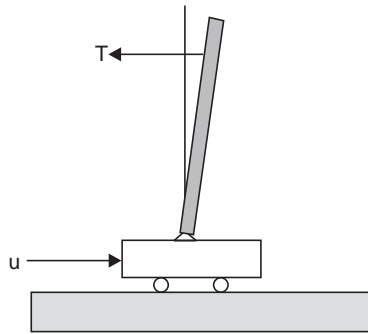



Fig. 13.15 Initial conditions $x_1(0) = 1^0$, $x_2(0) = 0$ dps, U* = 2.5

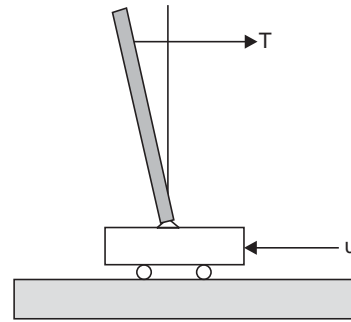



Fig. 13.16 Iteration 3 : $x_1(3) = 1.409^0$, $x_2(3) = -0.241$ dps, $u^* = -2.840$

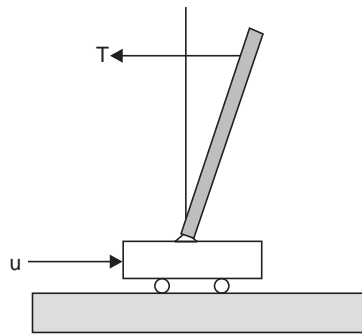



Fig. 13.17 Iteration 12 : $x_1(12) = 2.011^0$, $x_2(12) = -0.199$ dps, $u^* = 4.650$

13.9 DESIGN OF FUZZY LOGIC CONTROLLER IN SIMULINK AND MATLAB ENVIRONMENT

We shall now undertake the problem of designing a fuzzy logic controller in MATLAB and Simulink [57] environment for a closed loop control system. In order to put the reader on a familiar footing we shall consider first the design of PID controller by writing MATLAB script then simulate the dynamics using Simulink Library function blocks. A position control system employing an armature controlled DC servomotor will be used as illustrative example.

13.9.1 Iterative Design Procedure of a PID Controller In MATLAB Environment

We shall illustrate below, in a MATLAB environment, the iterative design procedure of a PID controller, followed by the design of a Fuzzy Controller, for controlling the position of a mechanical load driven by a DC servomotor.

Problem statement

Example 13.3 A PID controller is to be designed for a position control system, with unity feedback, employing an armature controlled DC motor to meet the following specifications :

Design specifications:

Steady state error in angular position, $\theta_{ess} = 0$

Settling time, $t_s = 0.2$ second

Overshoot ≤ 5 %

Model of armature controlled DC motor

The transfer function of the DC servomotor has been derived in the Appendix [see equation A.71] and is reproduced below:

$$\frac{\theta_m(s)}{E_a(s)} = G(s) = \frac{K_T}{JL_a s^3 + (R_a J + L_a B)\, s^2 + (R_a B + K_b K_T)\, s} \tag{13.27}$$

where J = Total moment of inertia of the rotor together with that of reflected load on the rotor side

B = Total viscous friction of the rotor together with that of reflected load on the rotor side

K_T = Motor torque constant

K_b = back emf constant

R_a = Armature resistance

L_a = Armature inductance

In an armature controlled DC motor, the armature inductance is negligibly small compared to armature resistance, in which case the transfer function in (13.27) can be written as

$$\frac{\theta_m(s)}{E_a(s)} = G(s) = \frac{(K_T/R_a J)}{s^2 + (B/J + K_b K_T/R_a J)s} \tag{13.28}$$

System parameters

Let us assume the following parameters for the position control system

$J = 5.0 \times 10^{-4}$ kg m^2/sec^2

$B = 6.0 \times 10^{-3}$ Nm/sec ;

$K_T = 0.085$ Nm

$K_b = 0.1$ V/rad/sec

$R_a = 5$ ohm

L_a = negligible

Step response of open-loop system

We write the following MATLAB script to simulate the open-loop response of this system to study its behavior to a step input.

```
% Script_Example13.3.m
J=5.0e – 4; B =6.0e – 3; KT = 0.085; Kb = 0.1; Ra = 5;
num =[(KT)/(Ra*J)];            % Numerator of transfer function in Equation (13.28)
den = [1 ((B/J) + (Kb*KT)/(Ra*J)) 0];       % Denominator of transfer function
g=tf (num, den);          % MATLAB function "tf" establishes the transfer function
t=0:0.001:0.2             % time period over which the response is required to be
                            computed and plotted
step(g,t);                % Step response with plot
```

The response of the rotor position will be found to grow continuously when a unit step voltage is applied to the armature terminals. The open-loop system is acting as a pure integrator. So, there is a need for a feedback that will stabilize the system for a step input.

The closed-loop transfer function of the system with unity feedback can be obtained by adding the two lines of code given below in the Script_Example13.3.m, followed by command to plot step response.

```
h =1                      % Unity feedback transfer function
gfb =feedback (g,h) ;     % Function "feedback" gives closed-loop transfer function
step (gfb, t) ;           % Step response of closed-loop system with plot;
```

From the closed-loop response, we find that the settling time is 1.55 second which is much higher than the desired value of 0.2 second, even though the overshoot is 0. In order to improve the settling time, we choose the control structure shown in Fig. 13.18, where the controller $G_c(s)$ has the transfer function of the form:

$$G_c(s) = K_p + \frac{K_i}{s} + K_d s = \frac{K_d s^2 + K_p s + K_i}{s}$$

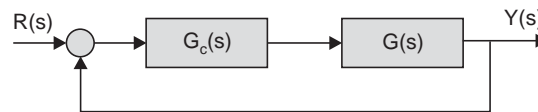



Fig. 13.18 PID control structure of position control system in Example 13.3

The objective in the controller design is to select the suitable values for the parameters for K_p , K_i , and K_d to satisfy specifications mentioned in the problem statement. It is to be mentioned here that in majority of cases a PI or PD controller may be adequate, and there may not be any need to use a PID controller. As we will find in this example, a PD controller is adequate to bring the system response to within acceptable limits of the prescribed values. However, full PID controller may be designed following similar trial and error procedure.

Initially, we set $K_i = K_d = 0$ and $K_p = 10$. The compensator transfer function, becomes $G_c(s) = K_p$. Now add the following lines to the MATLAB Script_Example 13.3.m.

```
% Add proportional control
Kp=10;                    % This first trial value chosen arbitrarily
%Kd = 0.5
 %Ki=0.01
 num1= [Kp];              % Numerator for proportional controller
%num1=[Kd Kp]             % Numerator for PD controller
% num1 = [ Kd Kp Ki]      % Numerator for PID controller
den1=[1];                 % Denominator for proportional and PD controller
% den1=[1 0];             % Denominator for PID controller
num2=conv (num1, num);
den2=conv(den1,den)
% Convolve the numerators and denominators of the controller and the plant
g1=tf (num2, den2);
t=0:0.005:1;              % Time interval for computing and plotting response
g2=feedback (g1, 1);      % Obtain the overall closed-loop transfer function
step (g2, t);             % Obtain the step response
```

We will find from the step response that the addition of a proportional gain gives rise to a large overshoot (23%) in the response and still does not meet the settling time criteria (0.457 second instead of 0.2 second). Now we set $K_d = 1$ as a first trial to arrest the large overshoot. The controller now becomes $G_c(s) = K_d s + K_P$. This change is incorporated in the script by modifying num1 as num1=[K_d K_p] and introducing $K_d = 1$ before the step response is obtained. The step response shows that the overshoot is zero but the settling time (0.29 second) is still much larger than the required value. In order to improve the settling time, value of Kp has to be increased. So we set $K_p = 15$ and keeping K_d unchanged, we compute the step response again, which reveals that overshoot = 0, settling time = 0.131 second, steady state error = 0 and rise time = 0.0731 second. Probably, we can afford to allow a little bit overshoot, so we decrease K_d to 0.75, keeping K_p unchanged at 15. The step response shows an overshoot of 1.81%, settling time, $t_s = 0.0997$, rise time $t_r = 0.0744$ and steady state error is zero.

Hence, a PD controller is adequate for this problem.

In order to study the effect of a PID controller, modify the right hand side of the assignment variable num1 as num1 = [K_d K_p K_i] and that of den1 as den1 = [1 0] in the Script_Example 13.3.m and set a small starting value for K_i in the range 0.1 to 10. The integral action is essential for a system with finite steady state error.

The iterative design procedure of a PID controller has been illustrated in MATLAB environment. As in many cases of system design, there is no unique solution to the design problem of a PID controller. However, the final choice of the PID parameters may be dictated by hardware constraints.

The basic steps involved in development of a PID controllers may be summarized as follows:

1. Compute the open-loop and closed-loop response of the plant to find out its shortcomings in the transient as well as in steady state conditions.

2. Increase proportional control K_p to improve the rise time.

3. Add derivative action K_d to decrease overshoot.

4. Add integral action K_i to eliminate/decrease the steady-state error.

5. Fine tune K_p , K_i , and K_d until a satisfactory response is obtained.

13.9.2 Simulation of System Dynamics in Simulink for PID Controller Design

We shall now describe the procedure of simulating the system dynamics in Simulink for undertaking iterative design procedure of PID controller as well as fuzzy logic controller.

Typing *simulink* in Matlab command window and pressing return key will open **Simulink Library browser** in a tree format. In order to create a new model, select the **New Model** on the Simulink Library browser's toolbar by clicking left mouse once on **File.** A new untitled model window opens. Click **Continuous** to see the components of continuous system simulation, including the transfer function block (**Transfer Fcn**) and **Derivative** block. With left mouse pressed drag the **Transfer Fcn** block to the new model window, renamed **DC motor** as shown in Fig. 13.19. For renaming a block, just click with left mouse and edit the block name. A block can also be duplicated by pressing **ctrl** key and dragging the existing block. Similarly drag one block of **Derivative,** two blocks of **Sum,** three block of **Gain** (found under node **Math Operations**), one **Step** block (found under node **Sources**) four **Scope** block and one **(simout) To Workspace** block (found under node **Sinks**) to the new model window.

Connect the blocks as shown in Fig. 13.19 and use appropriate names for the blocks for easy identification and reference. For connecting the output port of one block with the input port of another, just place the cursor over the output port of the first block and observe that the cursor changes to a cross hair. Press and hold down the left mouse button and drag the pointer to the desired input port of the target block. When you notice the cursor changes to a double cross, release the left mouse button. The port symbols are replaced by a connecting line with an arrow showing the direction of signal flow. The connecting line segments are either horizontal or vertical. A branch line can be drawn from an existing line to carry signal of equal strength to the input of another block. To draw the branch line, position the pointer to the desired starting point on the existing line, press **ctrl** key and drag the pointer with left mouse pressed to the input port of the target block. The entire line can be drawn in more than one step, just releasing the left mouse at any point terminates the line there and by placing the pointer at the end of the line and pressing the left mouse change direction as desired (by $\pm$90 degree or 0 degree) and draw another line segment. For deleting a line select the line by pressing left mouse press delete key. With pressed left mouse on a line one can drag the line to a desired location and release the left mouse.

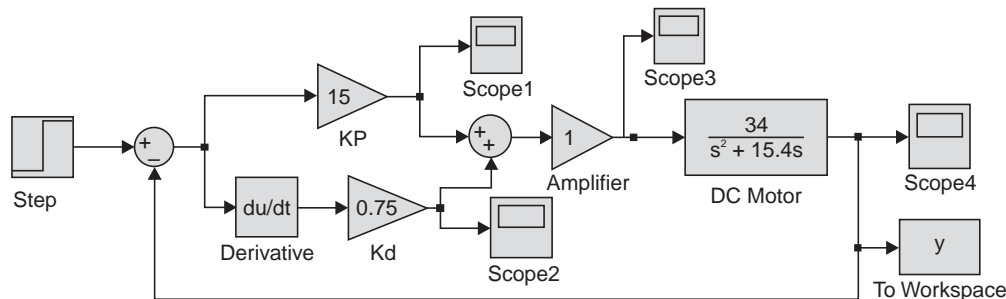



Fig. 13.19 Simulation diagram of a position control system using PID controller

Using the parameters of the armature controlled DC motor, the transfer function becomes

$$\frac{\theta_m(s)}{E_a(s)} = G(s) = \frac{34}{s^2 + 15.4s}$$

Double clicking the **DC motor** block opens the block parameters window; enter 34 in the window marked **Numerator** and [1 15.4 0] in the window marked **Denominator** and click **OK.** Double click the block **Step** and set **Step time** (time when the step input is applied) to 0 and accept the default values for **Initial** (= 0) and **Final** (= 1) for the step magnitude and save these by clicking OK. Similarly set the values for the blocks K*p* and K*d* and **Amplifier**. Double click the first **Sum** block and set the **List of signs** as [+ –] and that for the second as [+ +]. Open **simout** block by double clicking it and rename it as **y,** which stands for the output variable.

Save the model as **PID_Control.mdl** in **work** folder under MATLAB.

In order to start the simulation for solution, click **Simulation** in the open PID_control.mdl and click **Simulation parameters** to set **Start time** to 0 and **Stop time** to suitable value (1.5 sec in this run). Also click **Workspace I/O** and select **Save format** to **Array** then click OK for saving the parameters and closing the window. Now, click **Start** under **Simulation** menu and double click the scope to see the step response. Now iteratively change the settings of the controller parameters K*p* and K*d* for acceptable response. We can also plot the response in the MATLAB command window by typing >> plot(tout, y).

13.9.3 Simulation of System Dynamics In Simulink for Fuzzy Logic Controller Design

Let us now design a fuzzy logic controller for the same position control problem. We are interested to control the angular position of the rotor to align with some desired reference position. Let us define position error as $e = \theta_r - \theta_c$, where desired position θ_r is the set point. We can then use position **error** and rate of change of rotor error (**error-dot**) as the two input variables to the fuzzy logic controller and to generate the necessary voltage **u** to be applied to the armature of the DC motor.

The entire design will be undertaken in MATLAB (ver 6.5) and Simulink environment. The Simulink block diagram of the closed loop system incorporating a fuzzy logic controller is shown in Fig. 13.20. The gain elements A1, A2 and A3 in Fig. 13.20 are incorporated for adjusting scaling of the inputs and outputs of the fuzzy logic controller.

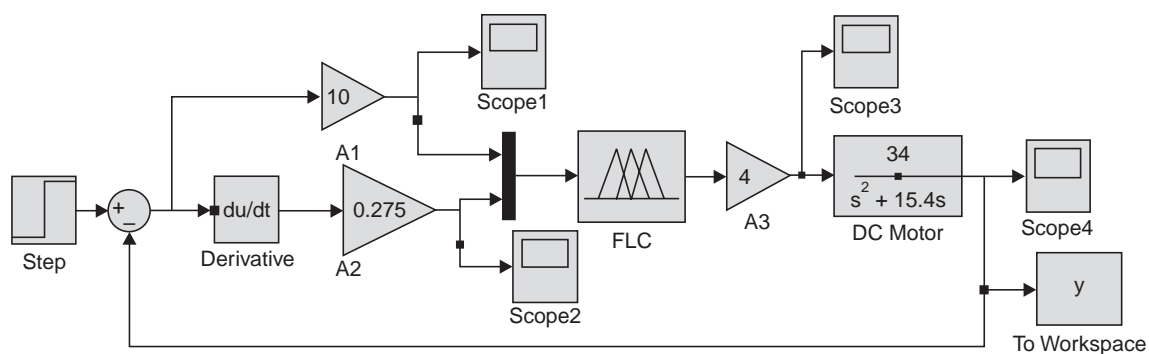



Fig. 13.20 Simulation diagram of position control system using fuzzy logic controller

The simulink model is created by following the procedure discussed in connection of the creation of **PID_ control.mdl** above. The block **Fuzzy Logic Controller** in the Simulink model is provided under **Fuzzy Tool Box** in the **Simulink Library Browser** window. Expand the tree by clicking the + sign in **Fuzzy Logic Toolbox.** You can see the **Fuzzy Logic Controller** block by placing the pointer over **Fuzzy Tool Box** and clicking the left mouse once. Drag the fuzzy logic controller block by pressing left mouse to the **New model** window. Rename the block as FLC, and double click the block and type **flc_servo** under **FIS File**. Save it by clicking OK. Similarly drag one block of **Mux** (found under node **Signal routing**) in the new model window. Save the untitled Simulink model as **FLC_control.mdl** in **work** folder under MATLAB.

Now enter fuzzy in MATLAB command window. It opens Graphic User Interface **FIS Editor** with default Untitled Mamdani Fuzzy Inference System (FIS). Click **File** menu and select **Sugeno** as a **new FIS** from drop down menu by clicking left mouse. This will enable us to build fuzzy logic controller with Sugeno Inference system (Vide Section 13.3). Since we have two input variables, **error** and **error-dot,** to the fuzzy controller and one output variable **u** for fuzzy logic controller, we have to introduce one more input variable to the **New FIS** model. Click **Edit** and click **Input** under dropdown menu **Add Variable.** One additional input membership box is added to the untitled **Sugeno FIS**. Now we shall assign names to the input and output variables. Left click the box labeled **input1** and type the name **error** replacing the text 'input1' in the **current variable** box located in the right bottom area of the editor window and press enter key. Similarly rename **input2** as **error_dot** and **output1** as **u**. Select **wtsum** (weighted sum) from **defuzzification** option and save FIS file by clicking **To disk** under

Export in the **File** menu as **flc_servo.fis,** the same name assigned to **FLC** block in the Simulink model **FLC_Control.mdl.**

We shall now make a choice about the type of membership functions and their numbers to cover the input and output domains. Let us decide to use trapezoidal membership functions for both the inputs and we take three membership functions to span both the input domains and take 5 membership functions to cover the output domain. We use the terms Z, P and N to describe both the inputs and terms NB, N Z P and PB for the output, where Z, N P, NB, PB stand for zero, negative, positive, negative big and positive big respectively.

Now select the input variable **error** by clicking left mouse and then click **Edit** and click **Membership Functions,** a window will open with default type and number of membership functions. Click **Remove All MF's** under **Edit** menu to clear the default membership functions for the input variable **error**. Now click **Add MF's** from Edit menu and select **trapmf** from **MF Type** drop down menu and **3** number of membership functions and click OK. Three trapezoidal membership functions labeled **mf1, mf2** and **mf3** will be displayed in the **Membership function plot** window. Change the input range from default value of [0 1] to [– 10 10] and press enter key to register the change. Select **mf1** by left mouse click and rename it to **N** and edit **params** [– 10 – 10 – 4 0] and press enter key. **Params** are the parameters [*a b c d*] of the trapezoidal membership functions (vide Section 12.7.1). Similarly, rename mf2 to **Z** and set parameters to [– 4 0 0 4] and mf3 to **P** with parameters [0 4 10 10]. Follow the same steps for error-dot. Now click the output variable u, Click **Remove All MF's** under **Edit** menu to clear the default membership functions for the output variable **u.** Click **Add MF's** from Edit menu and select **linear** from **MF Type** drop down menu and **5** number of membership functions and click OK. Change the range to [– 20 to 20] and press enter key. Now rename mf5 to **NB** and set params to [0.5 1 – 1].

It is to be noted that in the Tagaki-Sugeno type inference engine (vide section 13.3) the crisp output function u is given by $u = k\ f(x_1, x_2)$, where the function $f(x_1, x_2)$ in general, is nonlinear. For simplicity it is approximated by a linear function of the form $f(x_1, x_2) = k_o + k_1 x_1 + k_2 x_2$. In the parameter box **Params,** the linear coefficients are entered as [$k_2\ k_1\ k_o$].

Rename and set the parameters for the other membership functions in the output as shown below:

mf4 $\rightarrow$ **N**, params =[0.5 1 – 0.5]; **mf3** $\rightarrow$ **Z**, params =[0.5 1 0];

mf2 $\rightarrow$ P, params =[0.5 1 0.5]; **mf1** $\rightarrow$ **PB** , params =[0.5 1 1];

Note that the DC term ko in the membership functions **Z** is set to 0, since no DC voltage is required to be applied to the armature when the error in position and rate of change of position (error-dot) are approaching zero values. Also, note the sign of the constant terms k_0 for the output membership functions; it is **negative** for **NB** and **N** and **positive** for **P** and **PB.** Besides, the strengths of the coefficients k_1 and k_2 in the parameters are symmetric in **NB** and **PB**, **N** and **P** and the numerical values of these coefficients are chosen as a first trial and may have to be tuned during simulation to improve the system performance.

Now, we have to enter the rules for the fuzzy inference engine.

The rules for the fuzzy logic controller can be written from a knowledge of DC motor characteristics (vide Sec. A.9.1 in the Appendix), and the requirement of the control system (vide Section 13.1.3). We decide to use the following set of 9 *If-Then* rules:

1. If error is N and error-dot is P, then motor control voltage u is Z
2. If error is Z and error-dot is P, then motor control voltage u is P
3. If error is P and error-dot is P, then motor control voltage u is PB.

4. If error is N and error-dot is Z, then motor control voltage u is N
5. If error is Z and error-dot is Z, then motor control voltage u is Z.
6. If error is P and error-dot is Z, then motor control voltage u is P
7. If error is N and error-dot is N, then motor control voltage u is NB.
8. If error is Z and error-dot is N, then motor control voltage u is N.
9. If error is P and error-dot is N, then motor control voltage u is Z

For brevity, these rules are often shown as Fuzzy Associative Map (FAM) as in Table 13.4. The rule numbers are also shown in the table.

Table 13.4 FAM table

error \ *error-dot*	*P*	*Z*	*N*
P	PB (3)	P (6)	Z (9)
Z	P (2)	Z (5)	N (8)
N	Z (1)	N (4)	NB (7)

We can now enter the rules in the file flc_servo.fis. Click **Rules** under menu **Edit** in the FIS Editor : *flc_servo* window to open the **Rule Editor: flc_servo** window. In order to enter the first rule, sequentially select **N,** under variable **error**, by left mouse click, select **And** in the **connect** box and **P** under variable **error-dot** and **Z** under variable **output1** and then click **Add rule** located in the bottom area of the Rule Editor window. Immediately, the following text for rule 1 will appear in the display area.

1. If (error is N) and (error-dot is P) then (output1 is Z)(1)

The 1 in the parenthesis at the end of display is the weight or truth value assigned to the rule 1. This weight can vary from 0 to 1. Let us set the weight to 1,

In this way we enter all the rules in the FAM table. Any rule can be selected by left mouse click from the display area and edited as and when required.

Finally export the FIS file as flc_servo.fis in hard disk over writing the earlier file. Also export the file flc_servo.fis to work as a prerequisite to run the simulation of the FLC_Control.mdl.

We can see the firing of the rules in the flc_servo.fis file from FIS Editor by clicking **Rules** under **View.** We can see the effect of changing inputs **error** and **error-dot** by pressing left mouse on red vertical line and dragging left or right. The change in the controller output **u** will appear on the right most column of the **Rule Viewer**. Observation of the changes in the value of **u** in response to changes in error and error-dot on the rule viewer window will give a fair idea about the proper settings of the coefficients k_0, k_1 k_2 in the defuzzified output functions in the Sugeno-Tagaki inference engine. The control surface can be viewed by clicking **Surface** under **View**. By changing rules, one can see its effect on the control surface.

We can now open the model **flc_control.mdl** by double clicking it in the **work** folder and click **Start** in the open **flc_control**. Double click the scope and see the response. A typical response is shown in Fig. 13.21*b*.

The settings of the gain elements A1, A2 and A3 in Fig. 13.20 needs to be adjusted to get desired response. In addition to this, the membership functions of the inputs and outputs may

have to be tuned by trial and error to get desired response of the closed loop system. The set of parameters A1 = 10 A2 = 0.275, A3 = 4, Z = [0.5 1 0], P = [0.5 1 0.5], N = [0.5 1 – 0.5], PB = [0.5 1 1], Z = [0.5 1 – 1], along with the 9 rules in Table 13.4 will meet the specifications for the problem in Example 13.3.

The fuzzy logic controller will be found to be more robust compared to a PID controller. This is verified by considering the fuzzy logic controller tuned for one plant to a modified plant. Visualize a scenario that a robot arm is used to pick some objects of varying masses from the conveyor belt aided by a vision system, in a production line. The fuzzy logic controller has been designed and tuned for an expected average mass of the objects to meet prescribed overshoot in position and settling time in transient response. It is desirable, the performance should not degrade too much even if the masses change over wide ranges. We simulated the situation by multiplying the parameter J by a factor 0.5 and 2. In Figure. 13.21, we show the step response of the closed loop system incorporating PD and fuzzy logic controller for the case when J is multiplied by the factor of 2 where the respective controllers were tuned with nominal J. The step response with fuzzy logic controller. Figure. 13.21(*b*) shows very little overshoot whereas the response in Figure. 13.21(*a*) obtained with PD controller exhibits an overshoot of 11.7%.

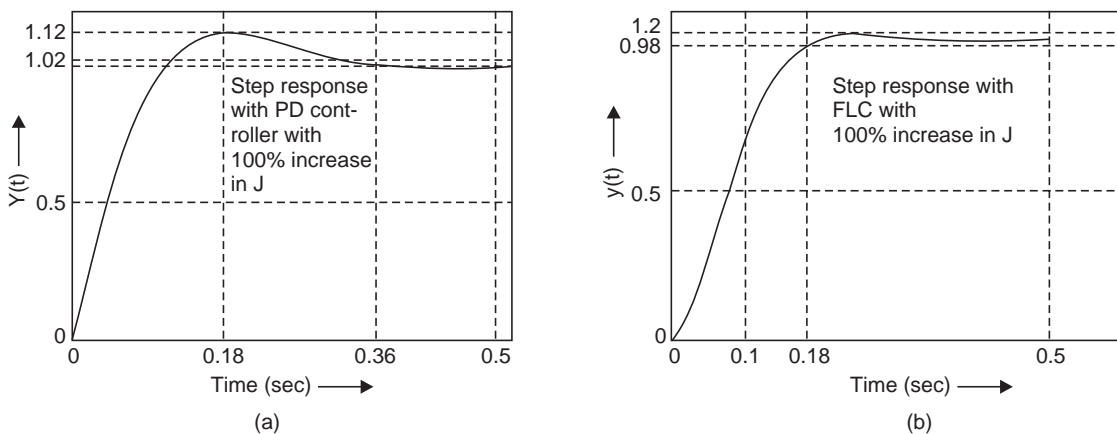



Fig. 13.21 Robustness test of PD and fuzzy logic controller, (a) step response of PD controller with changed G(s) given by G(s) = $17/(s^2 + 7.7s)$ (b) step response for fuzzy logic controller for the same G(s) used for PD controller.

We illustrated the design of fuzzy logic controller using the Tagaki_ Sugeno inference engine. Similar procedure may be followed for prototyping the fuzzy logic controller using Mamdani inference engine. In the design example we considered a very simple linear plant of a second order plant for illustrative purpose. The real power of fuzzy logic controller will be appreciated in the case of designing complex and nonlinear systems using the Simulink S-functions in MATLAB environment. The creation of user defined S-functions has been illustrated in Section 14.13.5 in Chapter 14.

PROBLEMS

P13.1 A position control system that positions a mass m has the following open-loop transfer function

$$\frac{Y(s)}{U(s)} = \frac{20}{ms^2 + 5s + 10}$$

Using the MATLAB and Simulink, do the following:

(*a*) For a unit step input, simulate the open-loop response of the system for m = 1 kg.

(*b*) It is desired to obtain a system response with settling time of less than 0.5 seconds and overshoot of less than 5%. Design a PID controller for the system using a mass m = 1 kg.

(*c*) Using the PID parameters obtained above, obtain the step response for the case when the mass of 1 kg is replaced by (*i*) m = 0.2 kg and (*ii*) m = 4 kg. Comment on system performance.

(*d*) Develop a fuzzy controller for the position control system tuned for m = 1 Kg. Obtain the system response with m changed to various values listed above and compare the result with that obtained with PID controller.

Comment on the performances of the two types of controllers.

P13.2 The open-loop plant transfer of an antenna positioning system is given by

$$\frac{Y(s)}{U(s)} = \frac{4}{s^3 + 5s^2 + 4s}$$

where the output is measured in *radians,* and the input is the applied torque in *Newton-meters* produced by the drive mechanism to rotate the antenna to the desired position.

(*a*) Obtain the unit step response of the closed loop system using the MATLAB and Simulink.

(*b*) Design a PID controller for the antenna positioning system with unity feedback control so as to get a critically damped response.

(*c*) Develop a fuzzy controller for the above system and compare the system performance with that obtained from the PID controller.

P13.3 The open-loop plant transfer function of a system is given by

$$G_p(s) = \frac{1}{s(s+\tau_1)(s+\tau_2)(s+\tau_3)}$$

(*a*) Show that the open-loop response will be unstable with $\tau_1 = 1$, $\tau_2 = 2$, and $\tau_3 = 3$, and the closed loop system with unity feedback is stable. Find the rise time, settling time and overshoot of the closed loop system with unit step input. Ans t_r = 8.91, t_s = 16.11, no overshoot.

(*b*) Develop a Simulink model of the system and obtain a satisfactory set of PID control parameters that will improve the settling time by at least 75% with a permissible overshoot of 5%.

Ans. t_r = 3.71, t_s = 10.6, overshoot = 4.7 % with K_p = 2.2, K_d = 1.0, K_i = 0

(c) Develop a fuzzy control model that can match the performance of the PID control system.

(*d*) Compare the performance of the PID and fuzzy controllers if the parameters τ_1, τ_2 and τ_3 vary ± 10% from their nominal values.

P13.4 Magnetic levitation is a technology presently used in high-speed trains in Japan. The idea is to levitate an object with a magnet and to propel the object at high speed.

It is desired to levitate a steel ball using the configuration shown in Fig. P13.4, where the ball position z is measured by using an optical sensor. The vertical position z of the steel ball of mass M is controlled by controlling the current input i to the electromagnet by varying a voltage V. The force F produced by the electromagnet must overcome the force Mg due to gravity, where g is the gravitational constant.

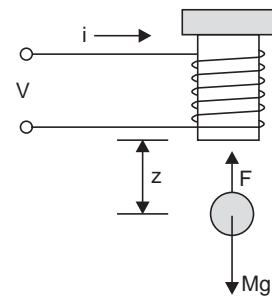
Fig. P13.4 Magnetic levitation

The mathematical model of this system can be obtained as follows:

for the steel ball we have

$$M\ddot{z} = Mg - F$$

and for the electrical circuit, using Kirchoff's voltage law, we have

$$L\frac{di}{dt} + Ri = V$$

where L and R are, respectively, the inductance and resistance of the coil. The coupling equation where the force F is related to the current i is given by

$$F = k_m \frac{i^2}{z^2}\text{, where } k_m \text{ is the proportional constant.}$$

Assume the following set of parameters for the levitation model under consideration

M	Mass of steel ball	25 milligrams (mg)
K_m	Magnetic constant	2.000×10^{-4} N(m/A)^2
R	Coil resistance	1.0 Ohms (Ω)
L	Coil inductance	0.02 millihenry (m H)
i	Coil current	[0, 3] Amps (A)
V	Coil voltage	[0, 5] Volts (V)
g	Gravitational constant	9.81 m/sec^2
z	Ball position	[min, max] =[2, 6] cm.

(*a*) Obtain a linearized model of the system at the minimum and maximum ball positions.

(*b*) For each linearized model, design a PID controller using a suitable set of criteria, and obtain simulation results of the system for various ball positions around the point of linearization. Is it possible to obtain a single linear model that will perform equally well?

(*c*) Develop a fuzzy controller that can magnetically levitate the steel ball for various positions within the range indicated. Simulate the system behavior for various disturbances that can act on the steel ball.

(*d*) Comment on the results obtained from the PID controller and the fuzzy controller.

P13.5 The longitudinal motion of an aircraft is represented by a set of linear differential equations as

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -0.1 & 1.0 & -0.02 \\ -7.8 & -0.05 & -5.5 \\ 0 & 0 & -8.0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} \delta_E$$

where

x_1 = angle of attack

x_2 = rate of change of pitch angle

x_3 = incremental elevator angle

δ_E = control input to the elevator actuator

(*a*) Obtain open-loop simulations results of the system with step, ramp, and sinusoidal inputs. What is your observation about the performance of the system to such inputs?

(*b*) Based on the knowledge gained from open-loop simulations, develop a fuzzy controller that will maintain stable aircraft performance in presence of unexpected disturbances.

P13.6 For the liquid level system shown in Fig. A.3 in Appendix A, the objective is to control the input flow such that desired heights of the fluid in the second tank can be maintained. Assume that the system has the following parameters:

$$C_1 = C_2 = 5 \text{ m}^2 \text{ and } R_1 = R_2 = 10 \text{ m/(m}^2\text{/sec)}$$

(*a*) Obtain simulations of the system using MATLAB and Simulink and find the response of the system with unity feedback. Also find the percent overshoot and settling time for a step input.

(*b*) Design a PID controller for controlling the input flow rate so as to maintain a desired height of the fluid in second tank with overshot less than 5% and the settling time found in (a) improves by at least 50% .

(*c*) Develop a fuzzy control system that can effectively maintain the liquid level heights in the second tank.

Discuss the performance of the fuzzy controller and that of the PID controller.

Hints: Using relations (A.15) and (A.17) in the Appendix, find the transfer function $H_2(s)/U(s)$ as :

$$G_1(s) = \frac{H_2(s)}{U(s)} = \frac{R_2}{R_1C_1R_2C_2s^2 + (R_1C_1 + R_2C_2 + R_2C_1)s + 1}$$

$$= \frac{10}{2500s^2 + 150s + 1}$$

With unity feedback control, the step response yields the following values:

Rise time, t_r = 23.4 sec, settling time (2% band) t_s = 126 sec, overshoot = 20.3%,

Using the PID parameter settings, $K_p = 1$, $K_d = 10$ and $K_i = 0$, the corresponding values are:

t_r = 26.8 sec, t_s = 76.3 sec, overshoot = 4.11%,

There is considerable improvement in settling time and over shoot by incorporating the PID controller.

CHAPTER

14

Nonlinear Systems: Describing Function and Phase-Plane Analysis

14.1 INTRODUCTION

In the Chapters 1 through 11, we confined our attention to the analysis and design of linear systems. This was not intended to imply that most control systems are linear in nature, rather, it was intended to gain some mathematical advantage in their analysis and design. Most of the control systems in use, on the contrary, are, to some extent, nonlinear in nature. We wanted to introduce the basic concepts of feedback control systems using linear systems with as little mathematical complexities as possible—the solution of linear differential equations can be obtained following a standard procedure and does not depend on the order of the system. The solution of nonlinear differential equations do not lend themselves to a common general approach-each nonlinear equation needs a distinct and complex approach (solutions of Bessel function, a second order differential equation, occupy volumes!).

Incidental and Intentional nonlinearities

Nonlinearities in the control system are of two general types: incidental and intentional. Some examples of **incidental** (or unintentional) nonlinearities are:

(*a*) Saturation
(*b*) Dead zone
(*c*) Hysteresis
(*d*) Backlash
(*e*) Nonlinear spring
(*f*) Nonlinear frictions: Static, coulomb etc
(*g*) Nonlinear fluid compressibility and flow rate.

The presence of such nonlinearities in the control system adversely affects system performance in general. Backlash, for instance, may cause instability in the closed loop system, and dead zone will cause steady-state error.

Intentional nonlinearities. Some nonlinear elements are intentionally introduced into a system to improve system performance or to simplify the implementation of the system, or both. An intentional nonlinear system when properly designed to perform a function is often superior from the points of view of cost, volume, weight and reliability to linear systems designed to perform the same task. The simplest example of such an intentional nonlinear element is a conventional relay. Another example is the nonlinear pitch or yaw dampers used in aircraft control systems and the nonlinear controllers of the actuating signal.

Other examples may be found in optimal control system where we often employ sophisticated nonlinear controllers. However, it should be borne in mind that even though intentional nonlinear elements may improve the system performance under certain operating conditions, they will, in general, degrade system performance under other operating conditions.

14.1.1 Some Phenomena Peculiar to Nonlinear Systems

The following facts and observations stands out as contrasting and peculiar phenomena of nonlinear systems alone for which there are no counterparts in linear systems [58-61].

(*a*) Dependence of output waveform on the magnitude of input signal
(*b*) Dependence of system stability on initial conditions
(*c*) Jump resonance
(*d*) Subharmonic and superharmonic oscillations
(*e*) Frequency quenching
(*f*) Frequency entrainment

(*a*) Dependence of output waveform on the magnitude of input signal

This is the most common experience of all engineers—when the input is a sinusoidal signal of appreciable amplitude to an electronic amplifier, the output signal is no longer a sinusoidal one. Besides, the output signal contains frequency components which were not present in the input. This is one of the reasons of degradation of music quality of your radio receiver or stereo amplifier at high volume—the music system generates some additional frequencies which were not present in the audio input—the vocal characteristic of your favorite artist.

(*b*) Dependence of autonomous system stability on initial conditions

We have studied the stability of linear systems in earlier chapters and observed that the stability of linear systems depends on the nature of the roots of the characteristic equation- if the system is stable it remains so, theoretically, for any initial conditions and it will produce bounded output when subjected to a bounded input. If the linear system is unstable, even with zero initial condition or input, the system will pick up a stray noise signal and the output amplitude will grow with time. This is not the case for a nonlinear system, which may behave as a stable system for some initial conditions and as unstable one for some other initial conditions (see Fig. 14.1(*a*) and (*b*)).

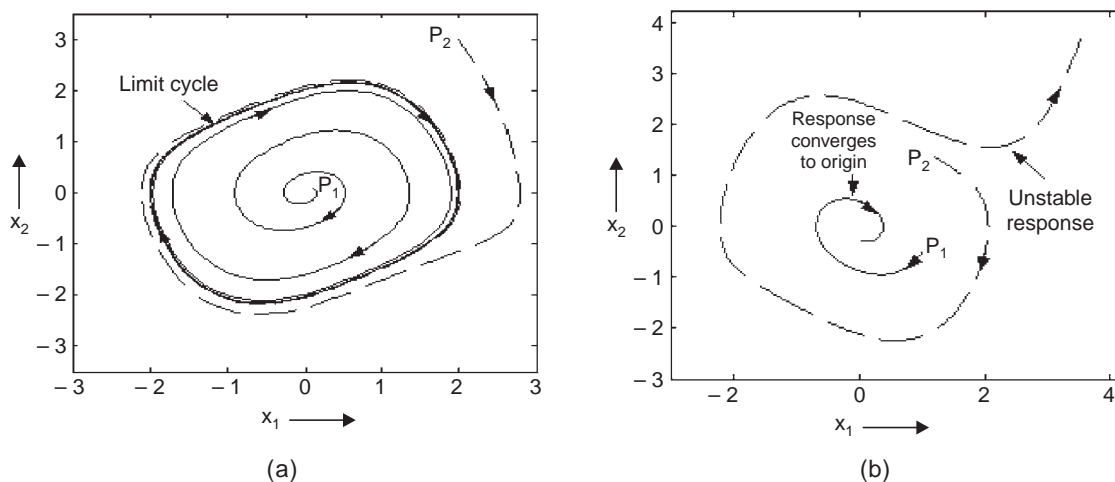



Fig.14.1 Dependence of response of nonlinear systems on initial conditions: (*a*) the response of $\ddot{x} - 0.4(1 - x^2)\dot{x} + x = 0$, converges to the limit cycle staring from initial conditions P_1 (0.1, 0.1) and P_2 (2, 3); (*b*) the response of $\ddot{x} + 0.4(1 - x^2)\dot{x} + x = 0$ converges to the origin starting from initial condition P_1(1.0, – 0.5) but diverges from initial condition P_2(1.2, 1.35). Here, the phase variables are chosen as state variables; $x_1 = x$, $x_2 = \dot{x} = \dot{x}_1$.

(*c*) Jump resonance

Let us consider the response of a mechanical system described by nonlinear equation of the form :

$$m\ddot{x} + b\dot{x} + kx + k_1x^3 = f \sin \omega t$$

where m = mass

x = displacement of mass

$kx + k_1x^3$ = nonlinear spring force

b = coefficient of viscous-friction and $f \sin \omega t$ is the forcing function.

The parameters m, k and b are positive constants, and k_1 may be either positive or negative.

If we vary the frequency ω of the forcing function by keeping its amplitude f constant and plot the magnitude of the displacement, we will get characteristics shown in Fig. 14.2(*a*) and (*b*) for $k_1 < 0$ and $k_1 > 0$ respectively. With reference to Fig. 14.2(*a*) suppose that the input frequency is such that the amplitude is equal to that corresponding to point A on the graph. As we increase the frequency, the amplitude X increases until point B is reached. If the frequency ω is increased further, there will be a jump in the characteristic from point B to point C, with associated change in amplitude and phase. This phenomenon is known as *jump resonance.* Any further increase in the frequency ω will cause the amplitude X to decrease monotonically from point C toward point E through D. Now, starting from high frequency corresponding to point E, if the frequency ω is decreased, the amplitude X will slowly increase through points E, D, C, until point F is reached. A further decrease in ω will cause another jump from point F to point G, associated with changes in amplitude and phase. Following this jump, any further decrease in frequency ω will cause the amplitude X to decrease monotonically following the curve from point G toward point A. The frequency response curves are discontinuous, and the change in amplitude is different for increasing and decreasing frequencies.

The output oscillations, between F and B cannot be observed experimentally. Similar jumps take place in the case of the system with $k_1 > 0$ as shown in Fig. 14.2(*b*). We thus see that for a given amplitude f of the forcing function there is a range of frequencies in which either of the two stable oscillations can occur. The phenomenon of jump resonance will occur when the damping term b is small and that the amplitude f of the forcing function is quite large so that the system operates with appreciable nonlinearity.

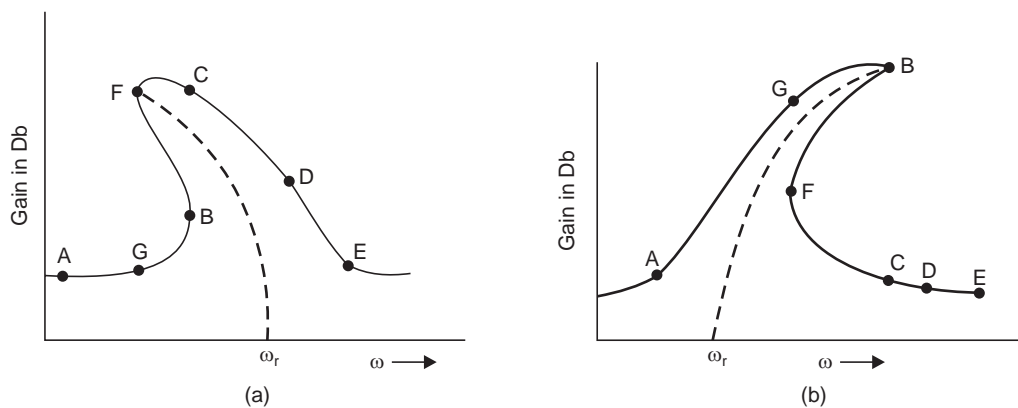



Fig. 14.2 Jump resonance in nonlinear system

Contrast this characteristic with the frequency response curve of a typical second order under-damped linear system as shown in Fig.14.3, where the gain is changing monotonically either for increasing or decreasing ω, without any jump change in gain.

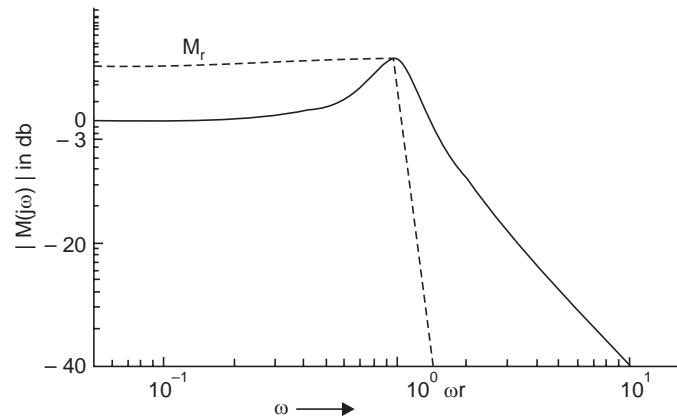



Fig. 14.3 Frequency response of an under-damped second order linear system

(*d*) Sub-harmonic and superharmonic oscillations

When a nonlinear system is driven by a sinusoidal signal of frequency ω, the output of the system may contain a periodic signal of frequency $\omega_h = m\omega$ or $\omega_s = \omega / m$, where $m > 1$ is an integer and is the order of super-harmonic/sub-harmonic oscillation. The steady state oscillation with frequency ω_h which is integral multiple of the forcing function frequency, is referred to as super-harmonic oscillation. And the steady state oscillation with frequency ω_s which is integral submultiple of the forcing function frequency, is referred to as sub-harmonic oscillation. Fig.14.4 shows sub-harmonic oscillation in the output along with the sinusoidal input.

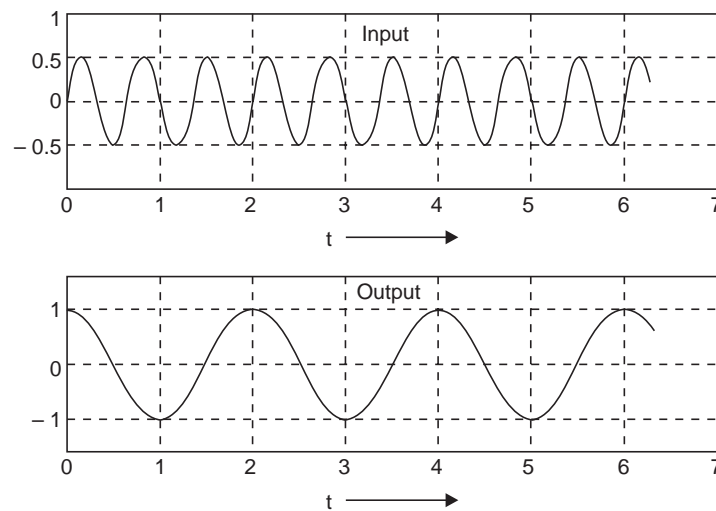



Fig. 14.4 Subharmonic oscillation in a nonlinear system

The generation of harmonic oscillations depend on system parameters and initial conditions, as well as on the amplitude and frequency of the forcing function. Sub-harmonic or

super-harmonic oscillations are not self-starting and to initiate such oscillations the system is subjected to some short of shock in the form of a sudden change in the amplitude or frequency of the forcing function. Once sub-harmonic oscillations are excited, they may be quite stable over certain ranges of frequency.

(*e*) Frequency quenching

When a nonlinear system is exhibiting a steady limit cycle oscillation of frequency ω_r, say, the limit cycle oscillation may be quenched by the application of another external sinusoidal signal with frequency ω_1, which is in no way related to ω_r. This phenomenon is known as *signal stabilization or asynchronous quenching.*

(*f*) Frequency entrainment

Another phenomenon which is peculiar to nonlinear system only is the *Frequency entrainment.*

Suppose a nonlinear system, which is susceptible to limit cycle oscillations with frequency ω_r, is subjected to sinusoidal excitation of frequency ω which is in the neighbourhood of ω_r, so that beats are produced. The beat frequency decreases linearly as the frequency ω approaches ω_r until a zone around ω_r is reached, when suddenly ω_r coalesce with ω as shown in Fig. 14.5. Beats are also produced in linear systems, and it can be shown theoretically and experimentally, that the beat frequency $|\omega - \omega_r|$ decreases continuously, as ω approaches ω_r shown by the dotted line in Fig. 14.5. The phenomenon of coalescing the limit cycle frequency in nonlinear system with the forcing function frequency in a zone is known as *frequency entrainment* and the band of frequencies in which this occurs is called *zone of frequency entrainment.*

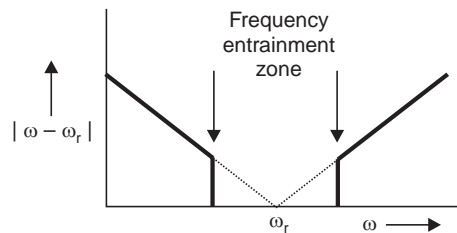



Fig. 14.5 The phenomenon of frequency entrainment

14.2 APPROACHES FOR ANALYSIS OF NONLINEAR SYSTEMS: LINEARIZATION

In a linear system the principle of superposition is valid, which is not the case for a nonlinear system. Therefore, the correlations between the frequency response and transient response developed for linear system, are not strictly applicable for nonlinear system. So, we need some methods for the analysis and design of practical nonlinear systems. Many simple systems where the inherent nonlinearity is not too severe, the nonlinearities are simply neglected and the system is analyzed as a linear system, producing acceptable results. This procedure is known as linearisation.

However, when a system or component contains a nonlinearity which can not be neglected, it may be analyzed by adopting one of the following approaches:

(*a*) The nonlinear element is replaced with one or more straight line segments.

(*b*) The system operation is restricted and small signal analysis is performed (*e.g.,* small signal analysis of electronic amplifiers).

(*c*) Numerical methods

(*d*) Graphical techniques: phase-plane analysis

(*e*) Describing function method of linearization

(*f*) Special technique of linearization for a class of nonlinear systems that can be represented in the form :

$$\overset{(n)}{x} = f(x, \dot{x}, \ldots \overset{(n-1)}{x}) + g(x, \dot{x}, \ldots\ldots \overset{(n-1)}{x})u \qquad (14.1)$$
$$y = x;$$

where $\overset{(n)}{x}$ is the nth derivative of x with respect to independent variable t, f and g are known bounded continuous nonlinear functions and $u \in R$ and $y \in R$ are the input and output of the system, respectively.

We shall consider only the Describing Function here followed by Phase plane analysis. But before we define describing function, a few words of the linearizing technique of the class of nonlinear system in Equation (14.1) are in order.

If the state vector $X = [x\,\dot{x} \ldots \overset{(n-1)}{x}]' \in R^n$ of the system in (14.1) is available for measurement and feedback then the system is controllable, for $g(X) \neq 0$ in certain controllability region $U_c \subset R^n$. It is assumed that $g(X) > 0$ without any loss of generality.

The control u can now be chosen to cancel the nonlinearity and its design can be completed based on linear control system theory like pole assignment. In particular, if the system output is required to track a model output y_m, the control law can be found as

$$u^* = \frac{1}{g(X)}[-f(X) + \overset{(n)}{y}_m + k'E] \qquad (14.2)$$

where $e = y_m - y = y_m - x$; and $k' = (k_1, \ldots\ldots, k_n)$ is the transpose of the feedback matrix k

and $E = [e\,\dot{e} \ldots \overset{(n-1)}{e}]'$ is the error vector.

With controller given by (14.2), the system (14.1) is now in closed loop configuration, and the tracking error of the model adaptive system is represented by the linear equation :

$$\overset{(n)}{e} + k_n \overset{(n-1)}{e} + \ldots k_1 e = 0 \qquad (14.3)$$

The tracking error can be forced to converge to the origin in the error space as $t \to \infty$ by placing poles of the characteristic Equation (14.3) in the left hand side of the s-plane.

14.3 DESCRIBING FUNCTION METHOD

The describing function method is an approximation method and is applied to those cases where the device characteristic deviates by a large amount from the linear operation.

Definition 14.1 The **describing function** N is defined [62] as the ratio of the fundamental component of the output y of the nonlinear element to the amplitude of the sinusoidal input x:

$$N = \frac{Y_1 \angle \phi}{X} \quad \text{when } x = X \sin \omega t \qquad (14.4)$$

where Y_1 is the amplitude of the fundamental component of the output y and ϕ is its phase shift, if any, with respect to the input.

14.4 PROCEDURE FOR COMPUTATION OF DESCRIBING FUNCTION

If a device is nonlinear and if a sinusoidal input is applied to this device, the output, in general, is not sinusoidal. The Fourier series for the output waveform of the device can be written as :

$$y(t) = A_0 + A_1 \cos \omega t + B_1 \sin \omega t + A_2 \cos 2\,\omega t + B_2 \sin \omega t + ... \quad (14.5)$$

The coefficients of this Fourier series are given by:

$$A_i = \frac{2}{\pi}\int_0^{\pi} y\,(t) \cos\,(i\theta)\, d\theta, \text{ where } \theta = \omega t \quad (14.6)$$

$$B_i = \frac{2}{\pi}\int_0^{\pi} y\,(t) \sin\,(i\theta)\, d\theta \quad (14.7)$$

Since, by definition, the describing function is concerned only with the fundamental frequency component of the output, the only coefficients of the Fourier series to be computed from relations (14.6) and (14.7) are for the value of $i = 1$. The fundamental component can be written as :

$$y_1(t) = A_1 \cos \omega t + B_1 \sin \omega t = Y_1 \sin (\omega t + \phi)$$

where $$Y_1 = (A_1^2 + B_1^2)^{1/2} \text{ and } \phi = \tan^{-1}\left(\frac{A_1}{B_1}\right) \quad (14.8)$$

The describing function, can, therefore, be found from the defining Equation (14.4) and using relations (14.6), (14.7) for $i = 1$ and Equation (14.8).

So the steps for finding describing function N are:

(*i*) Obtain the expression of $y(t)$ analytically or graphically by varying the amplitude of the input $x = X \sin \theta$.

(*ii*) Compute A_1 and B_1 from relations (14.6) and (14.7).

(*iii*) Compute Y_1 and ϕ from (14.8)

(*iv*) Then find N from (14.4)

It is apparent that the describing function N can be expressed as a complex quantity

$$N = |N|\, e^{j\phi} \quad (14.9)$$

where $|N| = \dfrac{Y_1}{X}$ is the magnitude and ϕ is the phase

The describing-function analysis of nonlinear systems is based on the following three assumptions:

(*a*) There is only one nonlinear element in the system. If there is more than one, the part of the system including all nonlinearities is considered as a single non-linear component for convenience. It is theoretically possible to consider systems with more than one isolated nonlinearity, but the analysis become unduly complicated except in very special nonlinear cases

(*b*) The nonlinear element is time-invariant.

(*c*) *If the input of nonlinear element n is a sinusoidal signal, only the fundamental component of the output of the nonlinearity contributes to the feedback signal in closed loop configuration.*

The last assumption is most important for the describing-function analysis. Its implication is explained with reference to the single-loop system in Fig. 14.6 consisting of a nonlinear element *n* and *a* plant G with transfer function G(*s*).

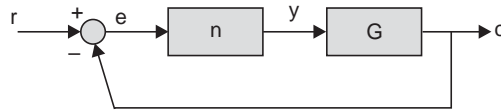



Fig. 14.6 Closed loop system with nonlinearity *n* and linear plant *G*

Let us assume that the input to the nonlinearity n is sinusoidal E $\sin \omega_1 t$, so that the output y of the nonlinearity is periodic in nature and can be expressed by the Fourier series in Equation (14.5) containing the fundamental frequency ω_1 of the input along with their harmonics. The dc component A_0 is, normally, zero for a nonlinear element with a characteristic which is symmetric in the first and third quadrant of the input-output space. Besides, if the coefficients of the harmonic components A_i, B_i , $i \geq 2$ are adequately attenuated while passing through the linear plant $G(j\omega)$, the output c contains essentially the fundamental frequency component. Therefore, the actuating signal $e(t)$ at the input of the nonlinearity, obtained by subtracting $c(t)$ from the reference input $r(t)$, also contains the fundamental frequency. The statement made in the assumption (c) above is a reasonable one for a suitable plant characteristic.

Now, if the transfer function $G(s)$ of the plant contain complex conjugate poles with damping ratio much less than 0.7, the frequency response will exhibit resonance peak at $\omega = \omega_r$ as shown in Fig. 14.7. If it so happens that the fundamental frequency of sinusoidal input to the nonlinearity is $\omega_1 = \omega_r/2$, then the amplitude of $|C(j\omega_1)|$ at the fundamental frequency ω_1 will be much less (see Fig. 14.7) than the amplitude at the harmonic frequency ω_r, thereby violating the assumption (c) of describing function. In such a case, the describing function analysis will lead to wrong conclusions.

The describing function method is not readily applicable to a system containing several nonlinear devices. The filtering may not be enough to ensure sinusoidal input to each of the nonlinear elements even if one is forced to be sinusoidal. Besides, the problems of handling two describing functions become intractable.

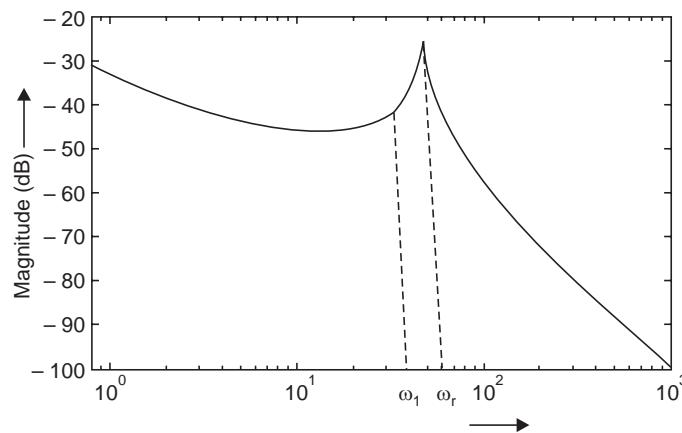



Fig. 14.7 Frequency response of a linear system with inadequate attenuation for $\omega \leq \omega_r$

14.5 DESCRIBING FUNCTION OF SOME TYPICAL NONLINEAR DEVICES

14.5.1 Describing Function of an Amplifying Device with Dead Zone and Saturation

The characteristic of the nonlinear element [59] is shown in Fig. 14.8(a). The characteristic reveals a dead zone for inputs from – d to + d and has a linear region of operation for inputs

between d and b or $-d$ and $-b$, and exhibits hard saturation for input amplitudes larger than b. In an actual device the saturation is not so sharp and more gradual, it has been assumed sharp for simplified description of the output. When the magnitude X is larger than b the periodic waveform of the output $y(t)$ of the nonlinear element is shown in Fig. 14.8(b). The output can be represented during the first half cycle as :

$$y(t) = \begin{cases} 0 & 0 < \omega t < \theta_1 \\ kX(\sin\theta - \sin\theta_1) & \theta_1 < \omega t < \theta_2 \\ kX(\sin\theta_2 - \sin\theta_1) & \theta_2 < \omega t < \pi - \theta_2 \\ kX(\sin\theta - \sin\theta_1) & \pi - \theta_2 < \omega t < \pi - \theta_1 \\ 0 & \pi - \theta_1 < \omega t < \pi \end{cases} \tag{14.10}$$

where k is the slope of the linear part and is found from Fig. 14.8(a) as $k = \dfrac{M}{b-d}$

It is to be noted that the angles θ_1 is the angle when the input equals the dead-zone amplitude and θ_2 is the angle when the device starts to saturate . Therefore, we can write

$$X\sin\theta_1 = d \text{ and } b = X\sin\theta_2, \text{ consequently } k = \frac{M}{X\sin\theta_2 - X\sin\theta_1} \tag{14.11}$$

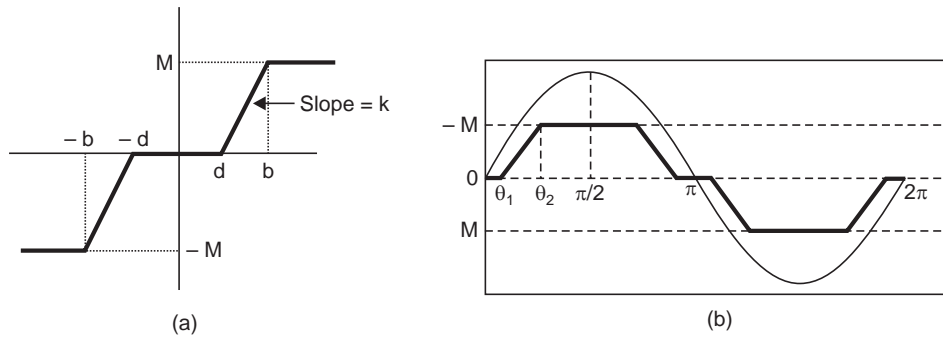



Fig. 14.8 Nonlinear characteristic with dead zone and saturation
(a) static (b) sinusoidal input and periodic output

If X lies between d and b, the output is not limited and θ_2 assumes the value of $\pi/2$.

The coefficients for the fundamental components can now be computed by integrating over the period 0 to π as follows:

$$B_1 = \frac{2}{\pi}\int_0^{\pi} y(t)\sin(\theta)\,d\theta$$

$$= \frac{2kX}{\pi}\left[\int_{\theta_1}^{\theta_2}(\sin\theta - \sin\theta_1)\sin\theta\,d\theta + \int_{\theta_2}^{\pi-\theta_2}(\sin\theta_2 - \sin\theta_1)\sin\theta\,d\theta + \int_{\pi-\theta_2}^{\pi-\theta_1}(\sin\theta - \sin\theta_1)\sin\theta\,d\theta\right] \tag{14.12a}$$

$$= \frac{kX}{\pi}\left[(\sin 2\theta_2 - \sin 2\theta_1) + 2(\theta_2 - \theta_1)\right] \tag{14.12b}$$

and $$A_1 = \frac{2}{\pi}\int_0^{\pi/2} y(t)\cos\theta\,d\theta$$

$$= \frac{2kX}{\pi}\left[\int_{\theta_1}^{\theta_2}(\sin\theta - \sin\theta_1)\cos\theta d\theta + \int_{\theta_2}^{\pi-\theta_2}(\sin\theta_2 - \sin\theta_1)\cos\theta d\theta + \int_{\pi-\theta_2}^{\pi-\theta_1}(\sin\theta - \sin\theta_1)\cos\theta d\theta\right] \quad (14.13)$$

$$= 0$$

Since A_1 is zero, there will be no phase shift of the fundamental component and the describing function N becomes a pure number. It is computed by using relation (14.8), (14.9) and (14.12*b*) as:

$$N = \frac{k}{\pi}\left[2(\theta_2 - \theta_1) + (\sin 2\theta_2 - \sin 2\theta_1)\right] \quad (14.14)$$

It can also be expressed in an alternative form as :

$$N = \frac{k}{\pi}\left[2(\theta_2 - \theta_1) + 4\sin\theta_1\cos\theta_2 - (\sin 2\theta_2 + \sin 2\theta_1)\right] + \frac{4M}{\pi X}\cos\theta_2 \quad (14.15)$$

The equivalence of (14.15) and (14.14) can be easily established by using the value of M from (14.11)

$$M = kX(\sin\theta_2 - \sin\theta_1).$$

Therefore, $4M\cos\theta_2 = 4kX(\sin\theta_2 - \sin\theta_1)\cos\theta_2 = 4kX(\sin\theta_2\cos\theta_2 - \sin\theta_1\cos\theta_2)$

So Equation (14.15) can be written as :

$$N = \frac{k}{\pi}\left[2(\theta_2 - \theta_1) + 4\sin\theta_1\cos\theta_2 - (\sin 2\theta_2 + \sin 2\theta_1) + 4\sin\theta_2\cos\theta_2 - 4\sin\theta_1\cos\theta_2\right]$$

$$= \frac{k}{\pi}\left[2(\theta_2 - \theta_1) - (\sin 2\theta_2 + \sin 2\theta_1) + 2\sin 2\theta_2\right], \text{ which yields Equation (14.14)}$$

The describing function can be expressed in terms of the dead zone d and saturation level b as a fraction of the input amplitude X by noting the following fact:

$$X\sin\theta_1 = d \quad \Rightarrow \quad \theta_1 = \sin^{-1}\left(\frac{d}{X}\right) = \sin^{-1} R_1 \quad (14.16a)$$

$$X\sin\theta_2 = b \quad \Rightarrow \quad \theta_2 = \sin^{-1}\left(\frac{b}{X}\right) = \sin^{-1} R_2 \quad (14.16b)$$

where $R_1 = d/X$, and $R_2 = b/X$ (14.16*c*)

Therefore, the describing function N in Equation (14.14) can be written, for X > b as:

$$N = \frac{k}{\pi}\left[2\left(\sin^{-1} R_2 - \sin^{-1} R_1\right) + 2R_2\sqrt{1 - R_2^2} - 2R_1\sqrt{1 - R_1^2}\right]\angle 0^\circ \quad (14.17)$$

The plot of N with X/b is shown in Fig. 14.9 with $b/d = 5$ and $b/d = 2$.

With reference to Fig. 14.8(*a*), we can find the describing function of (*i*) a saturation characteristic without any dead zone ($d = 0$) and (*ii*) a relay without hysteresis but with dead zone ($d = b$, hence $\theta_2 = \theta_1$) as special cases from the above expressions.

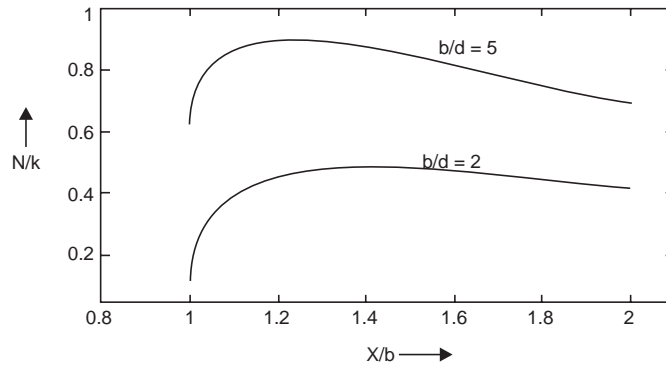



Fig.14.9 Variation of describing function, N with X/b for dead zone and saturation

14.5.2 Describing Function of a Device with Saturation but without any Dead Zone

We can draw the periodic waveform of the nonlinear element when excited with a sinusoidal signal as shown in Fig. 14.10 and proceed to compute the describing function N. Since, we have already computed the describing function of a more general nonlinear element shown in 14.8 of which the present device characteristic in Fig. 14.10 is a special case obtained with $d = 0$ such that $R_1 = 0$ and consequently $\theta_1 = 0$. Hence, for $X > b$, the expression for N in relation (14.15) can be recast with $\theta_2 = \theta$, and $R_2 = R = b/X$ as:

$$N = \frac{k}{\pi}\,[2\theta - \sin 2\theta] + \frac{4M}{\pi X}\cos\theta, \angle 0° = \frac{2k}{\pi}\left[\sin^{-1} R + R\sqrt{1 - R^2}\right], \angle 0° \qquad (14.18a)$$

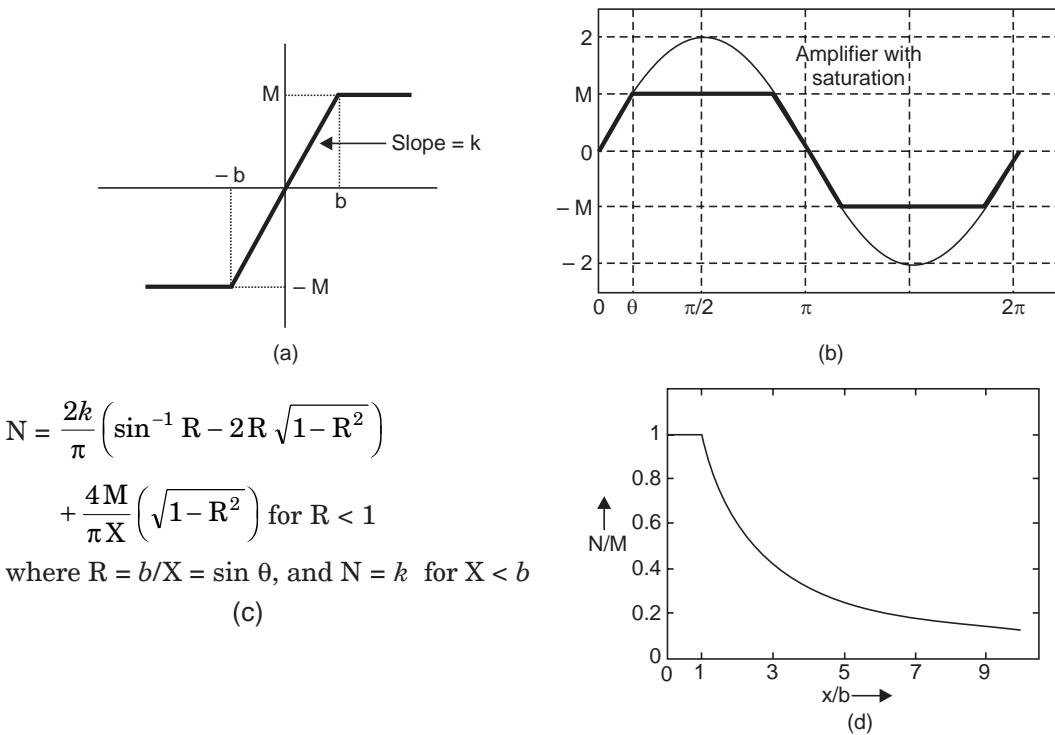



Fig. 14.10 Amplifier with saturation *(a)* static characteristic *(b)* sinusoidal input and periodic output *(c)* describing function and *(d)* plot of describing function

and when $X < b$ such that $\theta_2 = \theta = \pi/2$, the expression (14.18*a*) reduces to:

$$N = k, \angle 0° \tag{14.18b}$$

The variation of N with b/X is shown in Fig. 14.10(*d*).

14.5.3 Describing Function of a Relay with Dead Zone

The static characteristic of the nonlinear element is shown in Fig. 14.11(*a*) and its periodic output when subjected to a sinusoidal input is shown in Fig. 14.11(*b*). The describing function can be found by following the procedure given in section 14.5.1 and the result is shown in Fig. 14.11(*c*) and its magnitude variation with X/b is shown in part (*d*) of the Figure.

It can also be derived as a special case of the nonlinear characteristic in Fig. 14.8(*a*) with $b = d$, so that $R_1 = R_2 = R$, $\theta_2 = \theta_1 = \theta$, say. The describing function N for such an element is obtained from (14.15) as:

$$N = \frac{4M}{\pi X}\cos\theta = \frac{4M\sqrt{1-R^2}}{\pi X}\angle 0° \tag{14.19}$$

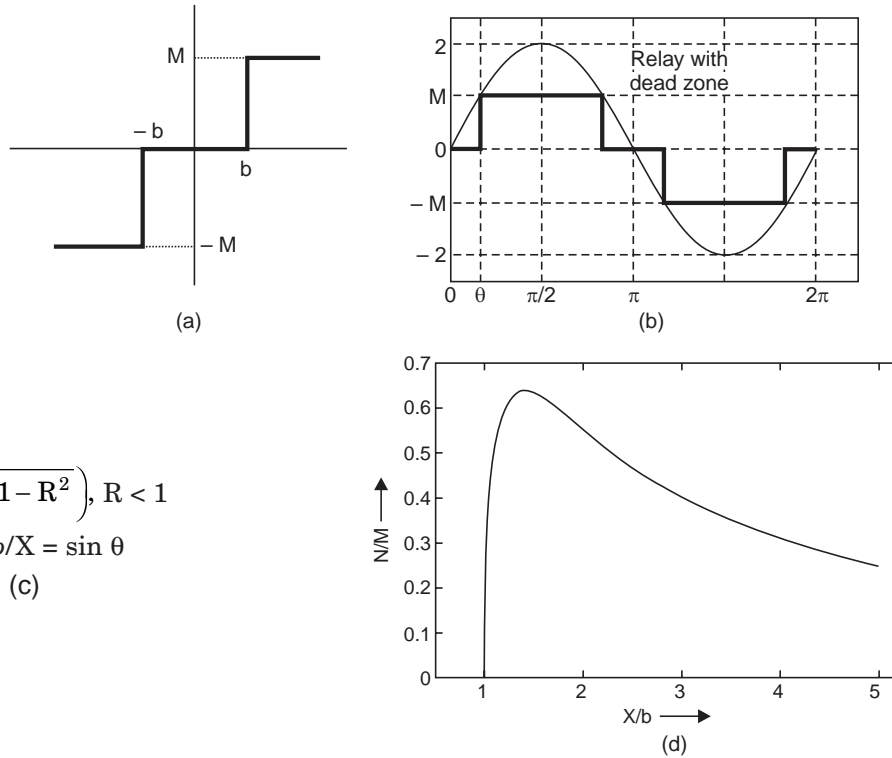



Fig. 14.11 Relay with dead zone (*a*) static characteristic (*b*) sinusoidal input and periodic output (*c*) describing function and (*d*) plot of describing function

14.5.4 Describing Function of a Relay with Dead Zone and Hysteresis

The static characteristic of the relay with hysteresis and dead zone is shown in Fig. 14.12(*a*) and the periodic waveform at the output of the nonlinearity can be obtained as shown in the Fig. 14.12(*b*). From the Fig. 14.12(*b*), the periodic output of the nonlinearity can be described as :

$$y(t) = \begin{cases} 0 & 0 < \omega t < \alpha \\ M & \alpha < \omega t < \pi - \beta \\ 0 & \pi - \beta < \omega t < \pi + \alpha \\ -M & \pi + \alpha < \omega t < 2\pi - \beta \\ 0 & 2\pi - \beta < \omega t < 2\pi \end{cases} \quad (14.20)$$

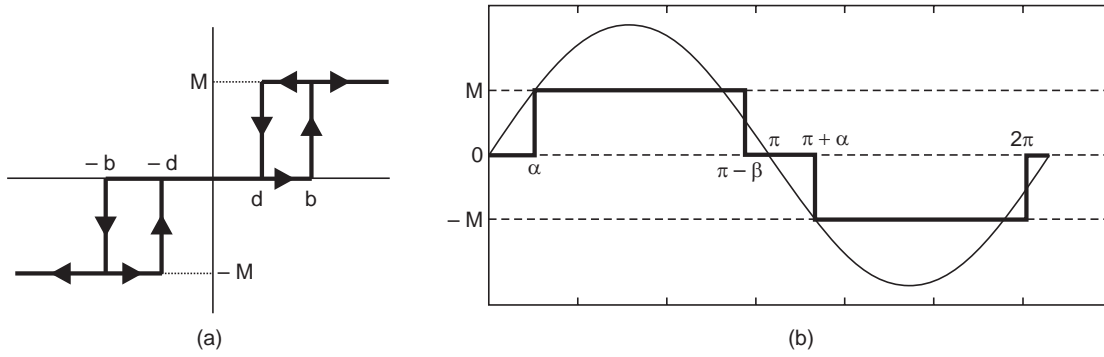



Fig. 14.12 Relay with dead zone and hysteresis (*a*) static characteristic (*b*) sinusoidal input and relay output

The fundamental components A_1 and B_1 can be found using relations (14.6) and (14.7) as:

$$A_1 = \frac{2}{\pi}\int_{\alpha}^{\pi-\beta} M\cos\theta\, d\theta = \frac{2M}{\pi}(\sin\beta - \sin\alpha) = \frac{2M}{\pi}(R_1 - R_2) \quad (14.21)$$

$$B_1 = \frac{2}{\pi}\int_{\alpha}^{\pi-\beta} M\sin\theta\, d\theta = \frac{2M}{\pi}(\cos\beta + \cos\alpha) = \frac{2M}{\pi}\left[\sqrt{1-R_1^2} + \sqrt{1-R_2^2}\right] \quad (14.22)$$

where $\sin\alpha = b/X = R_2$, and $\sin(\pi - \beta) = \sin\beta = d/X = R_1$

Here, we find that $A_1 \neq 0$, so, the describing function is a complex quantity with amplitude and phase shift. This is true for any 'memory' type nonlinearity, where the output is dependent on the time history of the input signal—it remembers whether the input is increasing or decreasing. Apart from the magnetization curve of the relay coil, a gear train with backlash is another kind of memory type nonlinearity. The describing function N of the relay with hysteresis and dead zone is computed with the help of relations (14.21), (14.22), (14.8) and (14.4). The variation of the magnitude and phase of the describing function are shown in Fig. 14.13(*a*) and 14.13(*b*) respectably.

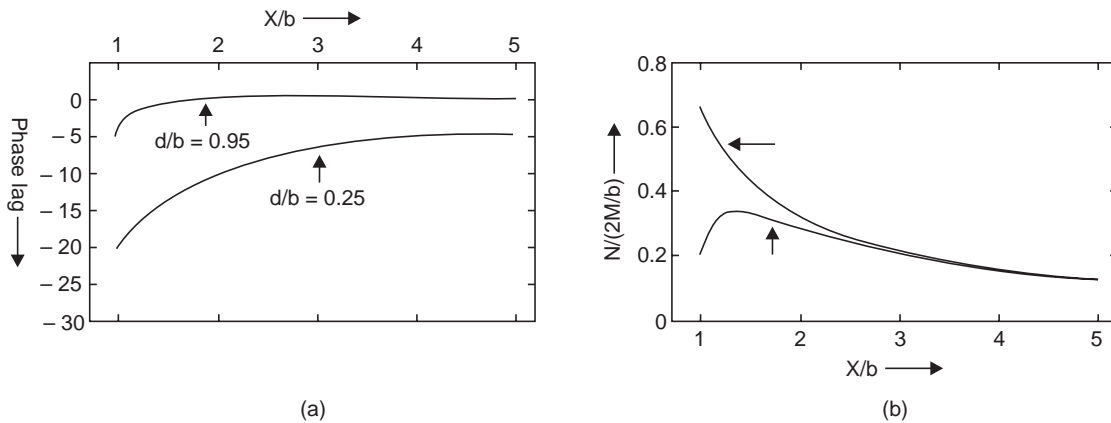



Fig. 14.13 The variation of (*a*) phase and (*b*) magnitude of the describing function N of a relay with hysteresis and dead zone with X/*b*

14.5.5 Describing Function of a Relay with Pure Hysteresis

The static characteristic of a relay with pure hysteresis is shown in Fig. 14.14(*a*). The sinusoidal input along with periodic waveform is shown in Fig. 14.14(*b*). The describing function N can be computed by following the procedure given in Section 14.5.1 and is given in Equation (14.23) below. The magnitude and phase plot of N is shown in Fig. 14.14(*c*) and 14.14(*d*) respectively. The phase lag is found to approach 90° at small amplitude of the input.

$$N = \frac{4M}{\pi X} \angle -\theta_1, \text{ where } \theta_1 = \sin^{-1}(b/X) \tag{14.23}$$

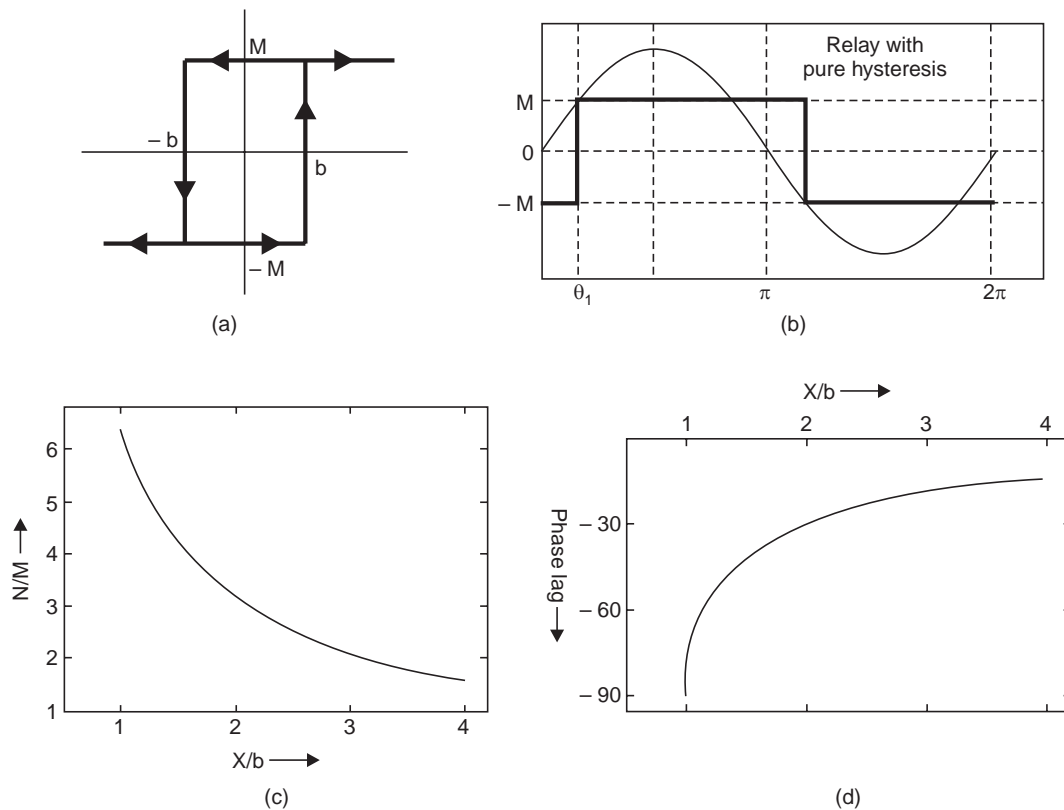



Fig. 14.14 Relay with pure hysteresis, (*a*) static characteristic (*b*) sinusoidal input with periodic output (*c*) plot of describing function magnitude N and its (*d*) phase with X/*b*

14.5.6 Describing Function of Backlash

The backlash characteristics are found in gear trains and mechanical linkages. The static input output characteristic of a backlash element is shown in Fig. 14.15(*a*). The characteristic is obtained by plotting movement of the input (driving) member along the x-axis and the movement of the output (driven) member along the y-axis. The characteristic can be explained with respect to the schematic diagram of Fig. 14.15(*b*). Assume that the driving member is initially placed at the center and is moving to the right hand (positive x) direction, because of the space d between the driving and driven member, the output member will remain stationary until the input member covers the distance d. Once the driving member is in contact with the driven member, it will start following the input member. When the input member

start moving in the opposite direction, the contact of the two members are broken and the output member remains stationary till the input member covers the distance $2d$ and the output member then start following input member.

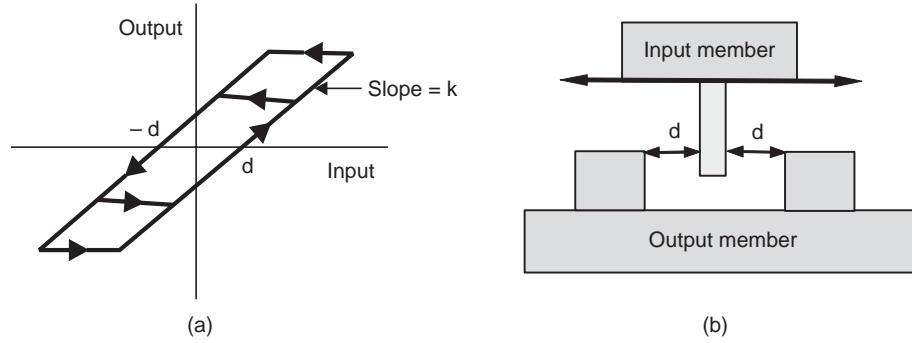



Fig. 14.15 Blacklash characteristic: (*a*) static (*b*) schematic diagram

Chapter 14

The sinusoidal input and the periodic output waveform in the steady state are shown in Fig. 14.16.

The output of the backlash element can be represented as follows:

$$y(t) = \begin{cases} k\,(\mathrm{X}\sin\theta - d) & 0 \le \theta \le \dfrac{\pi}{2} \\ k\,(\mathrm{X} - d) & \dfrac{\pi}{2} \le \theta \le \pi - \theta_1, \\ k\,(\mathrm{X}\sin\theta + d) & \pi - \theta_1 \le \theta \le \pi \end{cases} \quad \text{where } \sin(\pi - \theta_1) = \sin\theta_1 = \frac{\mathrm{X} - 2d}{\mathrm{X}}$$

The coefficients A_1 and B_1 of fundamental component are found as follows :

$$B_1 = \frac{2}{\pi}\left[\int_0^{\pi/2} k\,(X\sin\theta - d)\sin\theta d\theta + \int_{\pi/2}^{\pi-\theta_1} k\,(\mathrm{X} - d)\sin\theta d\theta + \int_{\pi-\theta_1}^{\pi} k\,(\mathrm{X}\sin\theta + d)\sin\theta d\theta\right]$$

We note that $\sin\theta_1 = \dfrac{\mathrm{X} - 2d}{\mathrm{X}} = 1 - 2\mathrm{R}$, where $\mathrm{R} = d/\mathrm{X}$ such that $d = \dfrac{\mathrm{X}}{2}(1 - \sin\theta_1)$ (14.24)

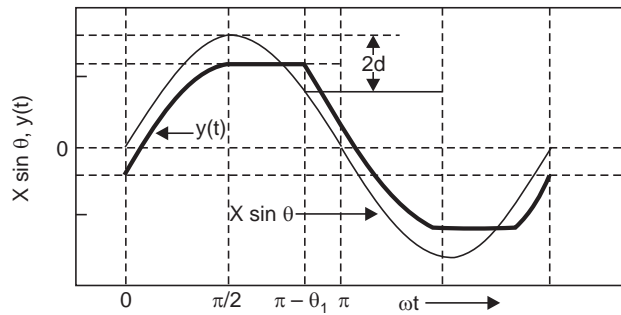



Fig. 14.16 Sinusoidal input and backlash output

Substituting this value of d and R, B_1 is computed as :

$$B_1 = \frac{kX}{\pi}\left[\frac{\pi}{2} + \theta_1 + \sin\theta_1\cos\theta_1\right] = \frac{kX}{\pi}\left[\frac{\pi}{2} + \sin^{-1}(1 - 2\mathrm{R}) + (1 - 2\mathrm{R})\,2\mathrm{R}\sqrt{\frac{1}{\mathrm{R}} - 1}\right] \quad (14.25)$$

Similarly, A_1 is given by:

$$A_1 = \frac{2}{\pi}\left[\int_0^{\pi/2} k\,(X \sin\theta - d)\cos\theta d\theta + \int_{\pi/2}^{\pi-\theta_1} k\,(X - d)\cos\theta d\theta + \int_{\pi-\theta_1}^{\pi} k\,(X\sin\theta + d)\cos\theta d\theta\right]$$

$$= -\frac{kX}{\pi}\cos^2\theta_1 = -\frac{4kX}{\pi}(1 - R)\,R \qquad (14.26)$$

The describing function N of the backlash element [63] will be a complex quantity and is computed with the help of relations (14.25), (14.26), (14.8) and (14.4). If the backlash is appreciable, the output member cannot move until the input member has moved through the backlash distance, so the output lags the input and the phase is negative. The phase lag approaches 0 as R = d/X approaches zero and approaches – 90° as R approaches unity, vide Fig. 14.17(a).

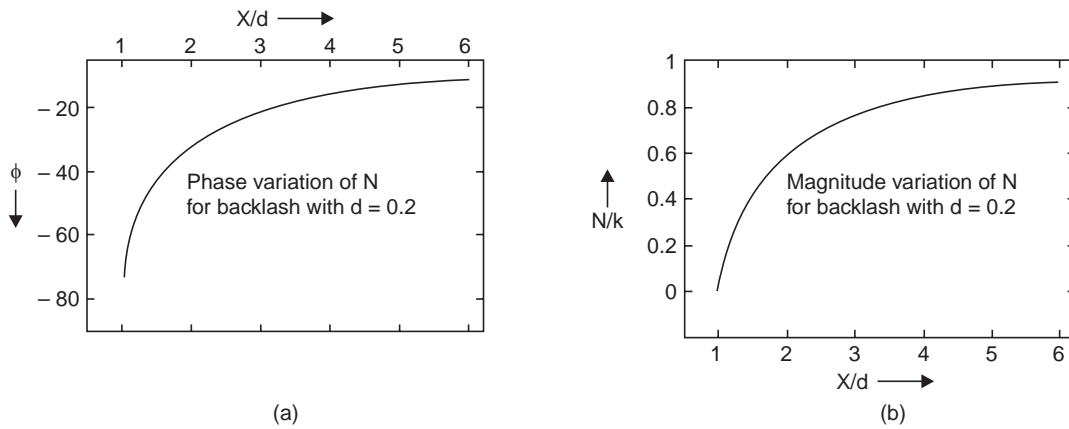



Fig. 14.17 (*a*) Variation of phase of N and (*b*) normalized magnitude N with X/*b* in a backlash

14.6 STABILITY ANALYSIS OF AN AUTONOMOUS CLOSED LOOP SYSTEM BY DESCRIBING FUNCTION

When the basic assumptions mentioned in Section 14.4 are satisfied, the sinusoidal response characteristic of the nonlinear element is expressed in terms of a describing function. It is to be noted that in deriving the describing function N, we used the periodic waveform of the nonlinear output in response to a sinusoidal input of a fixed frequency and N was found to depend only on the magnitude of sinusoidal input (and hence R). However, when the nonlinearity is an energy-storing element, the periodic waveform itself will depend on the input frequency ω as well as R. A simple saturation characteristic of an amplifier is described by a describing function N, which is dependent only on the amplitude of the input and independent of ω. On the otherhand, the describing function for a load consisting of inertia and damping depends not only on input amplitude, but also on the frequency of the input signal, because of the variation of the inertial effect with frequency. Therefore, in the most general case, the describing function is represented as N(jω), which is a complex quantity, with amplitude dependent on frequency as well as signal amplitude. The closed loop system with structure shown in Fig. 14.6 is now redrawn as shown in Fig. 14.18, where the nonlinear element is replaced by its describing function N(jω). We can now investigate the stability of the nonlinear system by slightly modifying some of the frequency domain methods employed for stability analysis of linear systems.

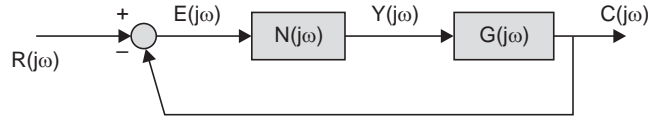



Fig. 14.18 Stability analysis using describing function

Since, the nonlinear element is treated as an element with a gain and phase varying with signal level and frequency, we can fix the amplitude X at a particular value and write the closed loop frequency response as :

$$\frac{C(j\omega)}{R(j\omega)} = \frac{N(j\omega)\,G(j\omega)}{1 + N(j\omega)\,G(j\omega)} \tag{14.27}$$

The characteristic equation, therefore, for a particular X can be written as :

$$1 + N(j\omega)\,G(j\omega) = 0 \qquad \text{or,} \qquad G(j\omega) = -\frac{1}{N(j\omega)} \tag{14.28}$$

The absolute and relative stability of the system may be analyzed by plotting the Nyquist diagram of G(*j*ω) in the gain and phase plane together with the plot of –1/N(*j*ω) on the same graph. Theoretically, N(*j*ω) is a function of both signal amplitude and frequency and –1/N(*j*ω) may be a family of curves. However, in most practical cases N(*j*ω) does not depend on ω, and for a given value of X it will be a single point on the graph and a single curve when X (or R) is varied over a range. We may have the following three possibilities regarding the plots of G(*j*ω) and –1/N(*j*ω) in the gain and phase plane.

(*i*) The two curves do not intersect and G(*j*ω) curve encloses (vide Nyquist plot in Chapter 6) all the curves –1/N(*j*ω).

(*ii*) The two curves do not intersect and G(*j*ω) curve do not enclose any of the curves – 1/N(*j*ω).

(*iii*) The curve G(*j*ω) intersects a plot of – 1/N(*j*ω).

For the case (*i*) above, the closed loop system is completely unstable, whereas for the case (*ii*) the closed loop system is stable. And the case (*iii*) is depicted in Fig. 14.19 and its implication is explained below.

For the purpose of the present discussion, we have considered a single curve of –1/N(*j*ω) which depends only on the amplitude X of the input signal and the direction of increasing X is also indicated ($X_5 > X_4 > X_3 > X_2 > X_1$) in the Fig. 14.19. From the Figure, we find that the two curves intersect at points $p_1(\omega_6, X_3)$ and $p_2(\omega_4, X_4)$ corresponding to angular frequencies ω_6 and ω_4 on G(*j*ω) curve and amplitudes X_3 and X_4 on –1/ N(*j*ω) curve. Suppose that the system is perturbed with noise of amplitude X lying between X_3 and X_4 so that the operation is confined between points p_1 and p_2. Since the part of the curve –1/N(*j*ω) between p_1 and p_2 is enclosed by G(*j*ω), the system is unstable and the amplitude will grow until it reaches the value X_4 corresponding to the point p_2. Any tendency of X to grow higher than X_4 will force the operation of the closed loop system to the stable zone. Consequently, the amplitude X will tend to decrease towards X_4. Any further decrease of X smaller than X_4 will make the system unstable and the amplitude will grow towards X_4. Therefore, the point p_2 is the point of stable operation. The closed loop system will exhibit a limit cycle with amplitude X_4 lying on the – 1/N(*j*ω) curve and frequency ω_4 corresponding to the point p_2 on the G(*j*ω) curve.

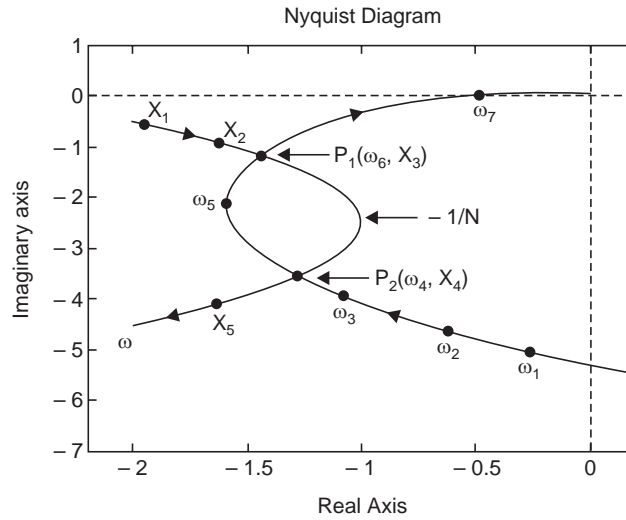



Fig. 14.19 Intersection of $-1/\,N(j\omega)$ curve with the Nyquist plot $G(j\omega)$

Now referring again to Fig. 14.19, if the system is initially perturbed with noise amplitude X_1 or X_2 which is much smaller than X_3, corresponding to point p_1, the closed loop system will operate in the stable region, since the part of the $-1/N(j\omega)$ curve for $X < X_3$ is not enclosed by the $G(j\omega)$ curve. So the value of X will decrease towards zero. If the amplitude slightly increases due to disturbance but still $X < X_3$, the system will remain in the stable zone of operation and the effect of disturbance will die out in course of time. If, on the other hand, the disturbance is of such magnitude that $X > X_3$, the system enters unstable region, further increasing the amplitude X. So moving along the increasing direction of X, the amplitude of X will grow until it reaches a stable amplitude X_4. Any further increase in X will move the operation from unstable region to stable region, having the effect of decreasing the value of X. So the point p_2 will become the point of steady limit cycle with amplitude and frequency corresponding to the point found from the N and G curve at p_2.

Example 14.1 With reference to Fig. 14.20, let $G(j\omega) = \dfrac{25}{s(s+25)}$, $H(j\omega) = \dfrac{1.28}{(s+5)}$ so that $G(j\omega)\,H(j\omega) = \dfrac{32}{s(s+5)\,(s+25)}$ and the nonlinear element is a relay with hysteresis and dead zone with $d = 0.02$ Amp, $b = 0.04$ Amp, M = 10 volts (see Fig. 14.12(*a*)). N is computed by using relations (14.4), (14.8), (14.21-14.22) and the plot of $-1/N$ is shown in Fig. 14.21. The $-1/N$ plot is superimposed on the Nyquist plot for $G(j\omega)\,H(j\omega)$ as shown in Fig. 14.22. It is found that the two plots intersect at (X_0, ω_0), which exhibits sustained oscillations with amplitude $X_0 = 0.12$ Amp at frequency $\omega_0 = 10.3$ rad/sec.

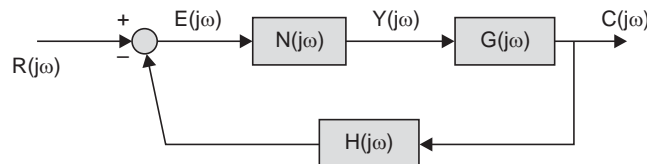



Fig. 14.20 Nonlinear system for example 14.1

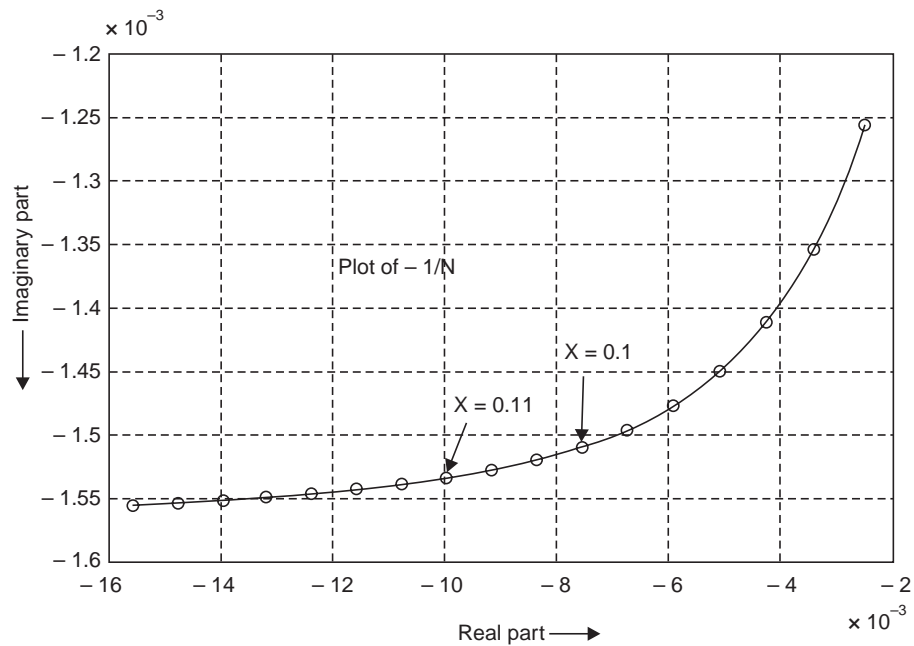



Fig. 14.21 Variation of –1/N with X/*b* for the relay in Example 14.1

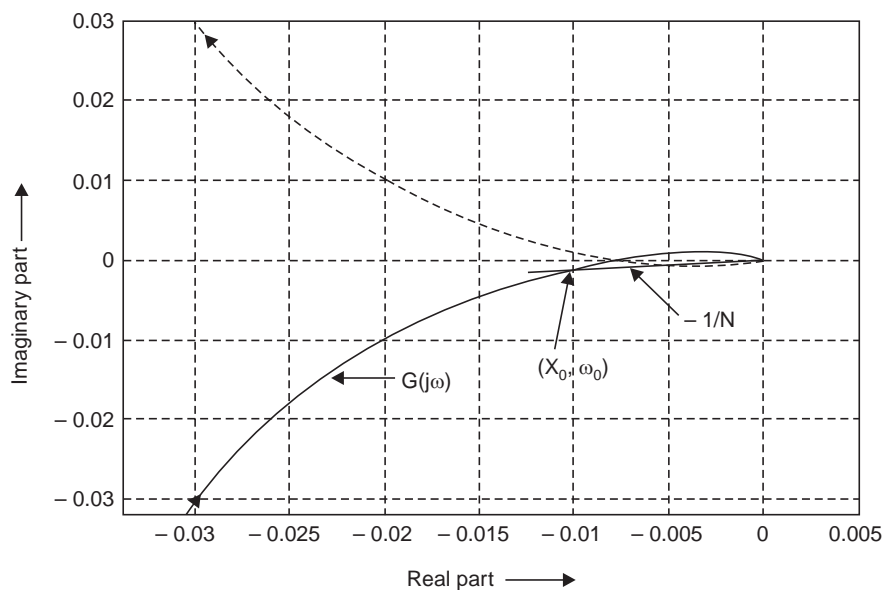



Fig. 14.22 Plot of G(jω) and –1/N for Example 14.1. The plots intersect at X_0, ω_0

14.7 GRAPHICAL ANALYSIS OF NONLINEAR SYSTEMS BY PHASE-PLANE METHODS

The phase (or state) plane method of analysis of nonlinear system [64] is a plot of the system trajectory in the plane defined by a system state and its time derivative. It derived its name

from its first application in studies of chemical equilibrium. In a closed-loop system, normally the output and its time derivative represent the phase plane. Alternatively, any other state, including the error and its derivative may be chosen to represent the phase plane.

The phase plane analysis is normally used for second-order systems even though it can be applied, in principle, to higher order systems. However, in a higher order system the phase variables and its time derivative can not be always identified with physical system variables and the method would lose its simplicity and convenience when applied to higher order systems. In the phase plane trajectory time does not appear explicitly, but can be recovered, when required, from the plot.

There are three popular methods for obtaining the phase plane plot when an analytic solution is difficult to obtain without the help of computers.

1. Solution of the equation by numerical methods. This method merely uses the phase plane for recording the data and interpreting the results of the solution.

2. The isocline method. Here the system equation is not solved, but rather the loci of constant slope (hence the name) on the phase plane are found; this permits the phase trajectory to be sketched.

3. Graphical construction methods, such as the delta method and Pell's method.

In the following sections, we present detailed procedures of constructing phase-plane plots and identify characteristic response patterns near common singularities.

14.8 PHASE-PLANE CONSTRUCTION BY THE ISOCLINE METHOD

In the isocline method of obtaining the phase plane plot, the system equation is rearranged to get the loci of constant slopes. The phase-plane trajectory can then be sketched with the help of constant slopes by starting from the initial conditions. The method is illustrated by considering a linear second order autonomous system given by

$$\frac{d^2x}{dt^2} + 2\delta\,\omega_n \frac{dx}{dt} + \omega_n^2 x = 0 \tag{14.29}$$

Let us normalize the time, so that the shape of the plot does not depend on ω_n.

Let $\tau = \omega_n t$, so that $d\tau = \omega_n\, dt$ (14.30)

Then Equation (14.29) becomes

$$\frac{d^2x}{d\tau^2} + 2\delta \frac{dx}{d\tau} + x = 0 \tag{14.31}$$

Let us define the following phase variables:

$$x_1 = x, \qquad x_2 = \frac{dx}{d\tau} \tag{14.32}$$

so that the Equation (14.31) can be written as :

$$\frac{dx_2}{d\tau} + 2\delta x_2 + x_1 = 0 \tag{14.33}$$

Dividing Equation (14.33) by x_2, we get

$$\frac{dx_2/d\tau}{dx_1/d\tau} = \frac{dx_2}{dx_1} = \frac{-x_1 - 2\delta x_2}{x_2} \tag{14.34}$$

Equation (14.34) is rearranged as

$$\frac{x_2}{x_1} = \frac{-1}{2\delta + dx_2/dx_1} = \frac{-1}{2\delta + m}, \text{ where } m = dx_2/dx_1 \tag{14.35}$$

For a given value of δ, loci of *constant slope* **m** or the isoclines of the phase trajectories may be plotted in the $x_1 - x_2$ plane. We show a number of isoclines in Fig. 14.23 after substituting various values for **m** in Equation (14.35) with δ = 0.4.

Starting from any initial conditions the phase trajectory may be determined using the isoclines. A typical trajectory is shown in Fig. 14.23, which is spiraling to the origin, indicating that the system is stable.

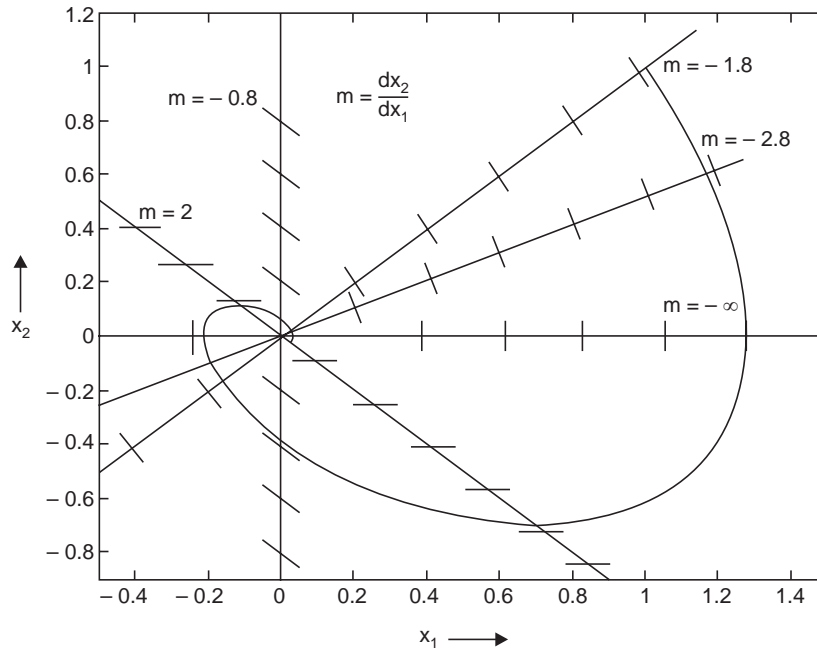



Fig. 14.23 Phase plane plot of Equation (14.35) using isoclines with δ = 0.4 and initial condition $\{x_1(0), x_2(0)\} = (1,1)$. The loci of constant slopes $dx_2/dx_1 = m$ for the phase plane plot are also indicated in the figure

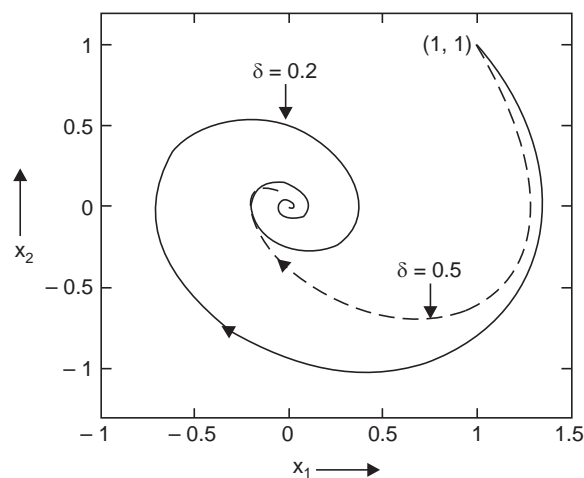



Fig. 14.24 The phase plane plot of Equation (14.35) with δ = 0.5 and δ = 0.2 starting from initial condition $\{x_1(0), x_2(0)\} = (1,1)$

If the time was not normalized, the shape of the phase plot will be elongated along the x_2 axis for $\omega > 1$ and shrinked for $\omega < 1$. The isoclines are also dependent on δ (see Fig. 14.24), and $\delta = 0$ will give rise to a circular trajectory corresponding to an oscillatory system. A negative δ will make the trajectory move away from the origin, implying an unstable system.

It is to be noted that the trajectory moves in the clockwise direction in the phase plane, since a positive $x_2 = dx_1/dt$ will result in a positive x_1 that increases with time whereas a negative x_2 will make x_1 to decrease with time.

Time is implicit in the phase plane plot and it increases along the trajectory. Time can be computed explicitly, when required, by noting the following fact.

We know, $x_2 = dx_1/d\tau$, therefore denoting the average value of x_2 by $\bar{x}_2$ over a small interval of time $\Delta\tau$, we can write

$$\bar{x}_2 = \frac{\Delta x_1}{\Delta \tau} \quad \text{or} \quad \Delta\tau = \frac{\Delta x_1}{\bar{x}_2} \tag{14.36}$$

where $\quad \Delta x_1 = x_1(\tau_1) - x_1(\tau_0) \quad$ and $\quad \Delta\tau = \tau_1 - \tau_0$

The time elapsed between τ_0 and τ_1 can be computed by using relation (14.36) as illustrated in Fig. 14.25. We can find $x_1(\tau_1)$ and $\bar{x}_2$ graphically as shown in Fig. 14.25 and the time interval $\Delta\tau$ can be computed from Equation (14.36). The process is repeated with successive Δx_1's. For acceptable accuracy, Δx_1 should be chosen properly such that the resulting Δx_2 is not very large.

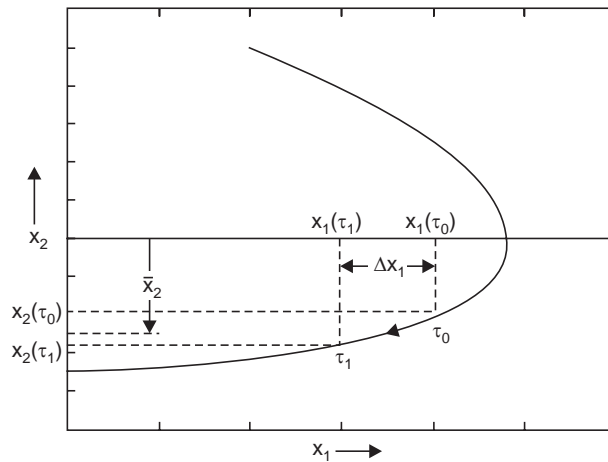



Fig. 14.25 Computation of time interval $\Delta\tau$ from phase plane plot. In the Figure $\bar{x}_2$ represents the average value of $x_2(\tau_1)$ and $x_2(\tau_0)$

14.9 PELL'S METHOD OF PHASE-TRAJECTORY CONSTRUCTION

For applying Pell's method [65] for plotting phase plane trajectory, the second order nonlinear equation is considered in its most general form:

$$\frac{d^2x}{dt^2} + \varphi\left(\frac{dx}{dt}\right) + f(x) = 0 \tag{14.37}$$

Now defining $x_2 = dx/dt = dx_1/dt$, the above equation can be rearranged, after dividing by x_2 to yield

$$\frac{dx_2/dt}{x_2} = \frac{x_2}{x_1} = \frac{-\varphi(x_2) - f(x_1)}{x_2} \tag{14.38}$$

The plots of $-\varphi(x_2)$ with x_2 and $-f(x_1)$ with x_1 are obtained as shown in Fig. 14.26. For a given initial condition $P[x_1(0), x_2(0)]$, the constructions of a segment of the trajectory are shown in the following steps :

1. Given the initial condition $P[x_1(0), x_2(0)]$ as shown in the Fig. 14.26, we drop perpendicular from P on the x_1 axis to get $OM = x_1(0)$ and extend the line PM to meet the curve $-f(x_1)$ at N such that $MN = f\{x_1(0)\}$. Now locate the point A on the x_1 axis such that MA = MN.

2. Similarly, for the given $P[x_1(0), x_2(0)]$, we draw PB perpendicular on the x_2 axis so that $OB = MP = x_2(0)$. Now extend PB to meet the $-\varphi(x_2)$ curve at C such that $BC = \varphi\{x_2(0)\}$. Extend the line segment MA on the x_1 axis upto point D such that AD = BC. Therefore, the segment of the x_1 axis between D and M equals the absolute values of the functions of x_1 and x_2. That is,

$$DM = DA + AM = \varphi\{x_2(0)\} + f\{x_1(0)\}$$

3. Join the points D and P. The slope of the line DP, denoted by m_1, is given by the tangent of the angle PDM, i.e.,

$$m_1 = \frac{PM}{DM} = \frac{x_2}{\varphi(x_2) + f(x_1)} \tag{14.39}$$

Now if the slope of the trajectory in Equation (14.38) is denoted by m_2 , then we observe that $m_1 m_2 = -1$. Therefore, the solution of the Equation (14.38) will move on a segment of a line, which is perpendicular to the line DP. So we draw a segment of line PQ which is perpendicular to the line DP to get the next point $Q(x_1(t_1), x_2(t_1))$ as shown in the Fig. 14.26. The process is repeated with Q as the new initial condition to get the complete trajectory.

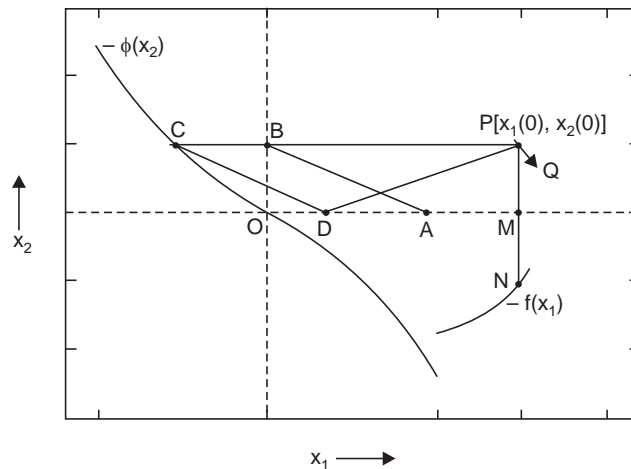



Fig. 14.26 Phase plane plot by Pell's method

The major advantages of this method are:

(*a*) The construction is very simple.

(*b*) It eliminates the trial and error approach of the delta method described below.

It is to be noted that plotting of the functions $\varphi(x_2)$ with x_2 and $f(x_1)$ with x_1, without the negative sign before the functions, would have served our purpose since we considered only the absolute values (line segments) of the functions for the construction of the triangle for a given initial condition.

Example 14.2 Consider a nonlinear closed loop control system employing pneumatic flapper nozzle and hydraulic valve, which is described by the following differential equation:

$$\frac{d^2x}{dt^2} + c_1\frac{dx}{dt} + \omega_1^2(1 + b^2x^2)\,x = 0 \tag{14.40}$$

We are interested to draw the phase plane trajectory by using Pell's method when the numerical coefficients are given by:

$$c_1 = 125,\ \omega_1^2 = 250000,\ b^2 = 0.60$$

We introduce time scaling as before : $\tau = \omega_1 t$ so that $d\tau = \omega_1 dt$ and the Equation (14.40) reduces to

$$\frac{d^2x}{d\tau^2} + \frac{c_1}{\omega_1}\frac{dx}{d\tau} + (1 + b^2x^2)x = 0 \tag{14.41}$$

Finally, after substituting the numerical values of the coefficients, the equation can be put in the form:

$$\frac{dx_2}{dx_1} = \frac{-0.25x_2 - (1 + 0.60x_1^2)\,x_1}{x_2} \tag{14.42}$$

Therefore, we identify $\varphi(x_2) = 0.25\ x_2$ and $f(x_1) = (1 + 0.60x_1^2)\ x_1$. The phase plane plot may now be obtained by Pell's method and is shown in Fig. 14.27.

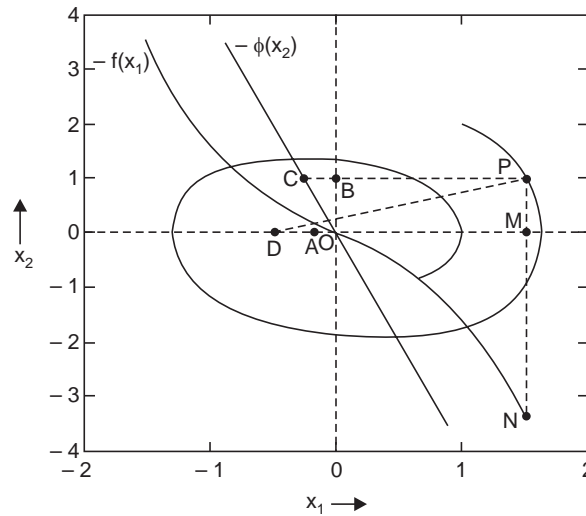



Fig. 14.27 Phase plane plot of the Equation(14.42) in Example 14.2

14.10 THE DELTA METHOD OF PHASE-TRAJECTORY CONSTRUCTION

The *delta method* [66] is also a popular method for the construction of phase plane plots of nonlinear systems, even though it is less accurate than that of Pell's Method. The delta method

approximates short segments of the phase plane trajectory with arc of circles. The first step of the method consists in expressing equation in following from:

$$\frac{dx_2}{dx_1} = \frac{-[x_1 + \delta(x_1, x_2)]}{x_2} \tag{14.43}$$

If δ is assumed constant in the neighbourhood of starting point (x_1, x_2) in the phase plane, then Equation (14.43) may be written as:

$$x_2\, dx_2 + (x_1 + \delta)\, dx_1 = 0$$

The solution of the above equation can be written as :

$$x_2^2 + (x_1 + \delta)^2 = r^2 \text{ , where } r \text{ is a constant}$$

which is the equation of a circle with center at $x_1 = -\delta$, $x_2 = 0$ and radius r.

Example14.3 Let us consider the problem of Example 14.2 and apply the delta method to obtain the phase plane plot. Therefore, the nonlinear system (14.42) may be arranged in the form :

$$\frac{dx_2}{dx_1} = \frac{-\{x_1 + (0.60\, x_1^3 + 0.25 x_2)\}}{x_2} \tag{14.44}$$

so that we can identify $\delta = 0.60x_1^3 + 0.25x_2 = \delta_a(x_1) + \delta_b(x_2)$, say

The steps of the delta method are now listed below:

1. Obtain a plot $-\delta_a(x_1)$ with x_1 and $-\delta_b(x_2)$ with x_2.

2. From an initial starting point, an arbitrarily small increment in x_1 is considered and the average of $-\delta_a(x_1)$ is found.

3. Take a trial increment in x_2 and estimate the value of $-\delta_b(x_2)$.

4. Draw an arc with center at $(-\delta, 0)$ through the initial starting point.

5. Correct the estimate of $-\delta_b(x_2)$ and repeat the steps 1 through 4 if necessary.

6. The steps 1 through 4 are repeated with new point found in step 5.

The trial-and-error procedure becomes necessary when δ depends on both the phase variables as in the present case.

14.11 CONSTRUCTION OF PHASE TRAJECTORIES FOR SYSTEM WITH FORCING FUNCTIONS

It is possible to construct phase plane plots of nonlinear closed loop system by using the procedures considered above. However, there will be considerable increase in complexity except for piece-wise-linear systems. The phase plane plots, can be obtained with CAD tools like MATLAB and SIMULINK.

14.12 SINGULAR POINTS

We have considered different methods for the construction of phase plane plots of nonlinear equations near the singularities in the previous sections. In order to examine the singularities [67] in detail, let us now consider the following pair of nonlinear state equations:

$$\begin{aligned} \dot{z}_1 &= a_{10} + a_{11}z_1 + a_{12}z_2 + P_1(z_1, z_2) \\ \dot{z}_2 &= a_{20} + a_{21}z_1 + a_{22}z_2 + Q_1(z_1, z_2) \end{aligned} \tag{14.45}$$

where P_1 and Q_1 contain only terms in z_1, z_2, and z_1z_2 of the second and higher degrees. The singular points of the above set of equations is the point where $\dot{z}_1$ and $\dot{z}_2$ are simultaneously

zero. Now according to Lyapunov, if the system is structurally stable, the nonlinear terms P_1 and Q_1 may be neglected compared to the linear terms near the singularities. A system is said to be *structurally stable* if a small deviation of its parameters does not change the character of the response. For example, a linear system with a pair of roots on the imaginary axis is not structurally stable.

Therefore, in a structurally stable system, first-order linear approximation of the function yields meaningful results around the singular points.

Neglecting the nonlinear terms P_1 and Q_1 in Equation (14.45) and taking the ratio we get the following expression:

$$\frac{\dot{z}_2}{\dot{z}_1} = \frac{dz_2}{dz_1} = \frac{a_{20} + a_{21}z_1 + a_{22}z_2}{a_{10} + a_{11}z_1 + a_{12}z_2} \tag{14.46}$$

We notice that the slope of the phase plane trajectory is indeterminate ($dz_2/dz_1 = 0/0$) at the singular point. The singularity of the Equation (14.46) is called a first-order singularity or simple singularity, whereas if the lowest terms present in Equation (14.46) were of n-th-order, it would be called an n-th-order singularity.

By using incremental variables as state variables (which is equivalent to coordinate translation), the constant terms in the Equation (14.45) may be eliminated, so that in the neighbourhood of singular point, we may write :

$$\begin{aligned} \dot{x}_1 &= a_{11}x_1 + a_{12}x_2 \\ \dot{x}_2 &= a_{21}x_1 + a_{22}x_2 \end{aligned} \tag{14.47}$$

By the translation, the origin of the new coordinate system is placed on the singular point.

If the eigenvalues of the system in Equation (14.47) are distinct, a similarity transformation (vide Section 2.12 in Chapter 2) may be used to represent the state equation in the decoupled form as:

$$\begin{aligned} \dot{y}_1 &= \lambda_1 y_1 \\ \dot{y}_2 &= \lambda_2 y_2 \end{aligned} \tag{14.48}$$

where λ_1 and λ_2 are the distinct eigenvalues given by :

$$\lambda_1, \lambda_2 = \frac{1}{2}\left[(a_{11} + a_{22}) \pm \sqrt{(a_{11} + a_{22})^2 + 4(a_{12}a_{21} - a_{11}a_{22})}\right] \tag{14.49}$$

Therefore, from Equation (14.48), we have

$$\frac{dy_2}{dy_1} = \frac{\lambda_2}{\lambda_1}\frac{y_2}{y_1} \qquad \text{or} \qquad \frac{dy_2}{y_2} = \frac{\lambda_2}{\lambda_1}\frac{dy_1}{y_1} \tag{14.50}$$

The Equation (14.50) has the solution

$$\ln y_2 = \frac{\lambda_2}{\lambda_1}\ln y_1 + \ln C \qquad \text{or} \qquad y_2 = C(y_1)^{\frac{\lambda_2}{\lambda_1}} \tag{14.51}$$

The roots or eigenvalues λ_1 and λ_2 in Equation (14.49) can be real, either positive or negative or can be complex conjugates with either positive or negative real parts depending on the parameter values. The various possibilities are examined below around the singular points. The parametric form of representation of x_2 or x_1 is of interest here and not their exact solution.

(*a*) Case of two real positive roots: $0 < \lambda_2 < \lambda_1$ and $0 < \lambda_2/\lambda_1 < 1$

We consider first the case of two real positive roots or eigenvalues. The plot of Equation (14.51) with $0 < \lambda_2 < \lambda_1$ for various initial conditions is shown in Fig. 14.28. This form represents

an unstable node. It may be shown that all the curves, with the possible exception of the degenerate case that lies on the y_1 axis, leave the singularity with the same slope. It is to be remembered that if x_1 and x_2 in Equation (14.47) represent position and velocity, respectively, after similarity transformation y_1 and y_2 are related to but not identical to position and velocity, respectively.

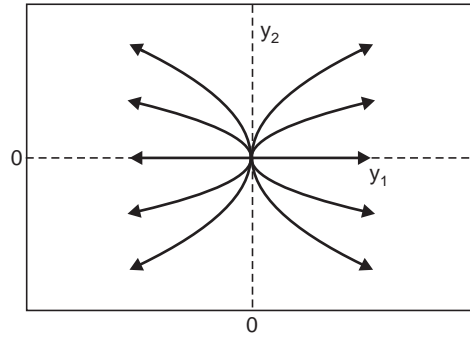



Fig. 14.28 The singular point is an unstable node shown in the state-plane defined by canonical variables. The eigenvalues satisfy the conditions:

$0 < \lambda_2 < \lambda_1$ and $0 < \lambda_2/\lambda_1 < 1$

(*b*) Case of two real negative roots $\lambda_2 < \lambda_1 < 0$ and $1 < \lambda_2/\lambda_1 < \infty$

If both roots were negative and real, the solution would have the same form as Equation (14.51), with the exception that $1 < \lambda_2/\lambda_1 < \infty$. Fig. 14.29(*a*) and (*b*) illustrate the form of the solution. The node in this case is stable. It can be shown that the trajectories approach the origin asymptotically to the y_1 axis and they are parallel to the y_2 axis at a large distance from the origin when the eigenvalues are widely separated [vide Fig. 14.29(*b*)]. The axes themselves are also part of trajectories under certain initial conditions. If we go back to the phase plane defined by x_1 and x_2 for the second order equation (14.47), the solution is plotted in Fig. 14.30 where the *eigenvectors* $y_1 = 0$ and $y_2 = 0$ are straight lines and become part of the trajectories. For condition $\lambda_2 < \lambda_1 < 0$, the eigenvector $y_1 = 0$ is referred to as slow eigenvector whereas and $y_2 = 0$ is referred to as fast eigenvector because of the relative phase velocities of the state points when moving along them.

(a)

(b)

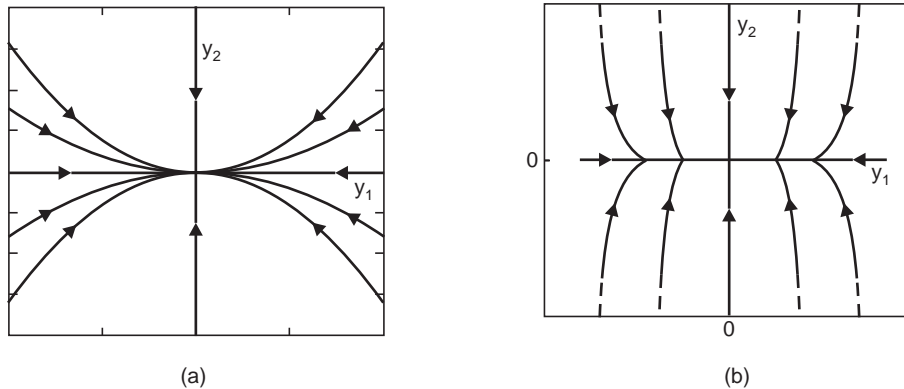



Fig.14.29. The singular point is a stable node. The plot is in the state-plane defined by canonical variables. The eigenvalues satisfy the conditions: $\lambda_2 < \lambda_2 < 0$ and $1 < \lambda_2/\lambda_1 < \infty$. Plot (*a*) when magnitudes of λ_1 and λ_2 are close to each other and plot (*b*) when they are widely separated

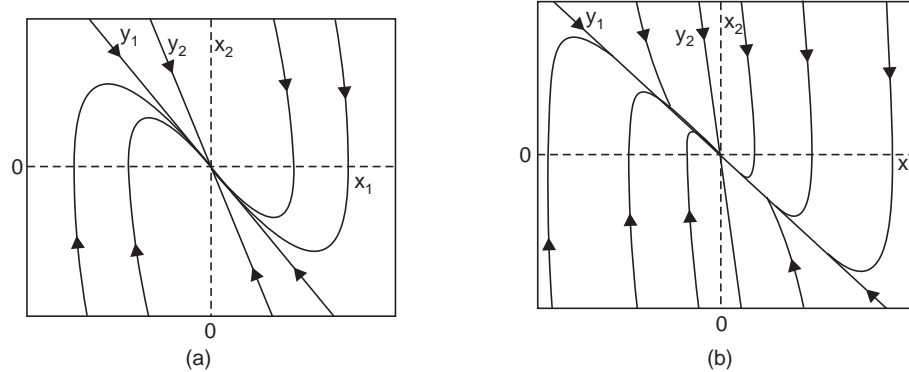



Fig. 14.30 The stable node when plotted in the state-plane defined by the physical variables. The eigenvalues satisfy the same conditions as in the case of Fig. 14.29, i.e., $\lambda_2 < \lambda_1 < 0$ and $1 < \lambda_2/\lambda_1 < \infty$, Plot (*a*) when magnitudes of λ_1 and λ_2 are close to each other and plot (*b*) when they are widely separated. $y_1 = 0$ is the slow eigenvector and $y_2 = 0$ is the fast eigenvector

(*c*) Case of one real positive and one real negative root

If one of the roots is real and negative and the other is real and positive, the resulting plot is called a saddle as shown in Fig. 14.31.

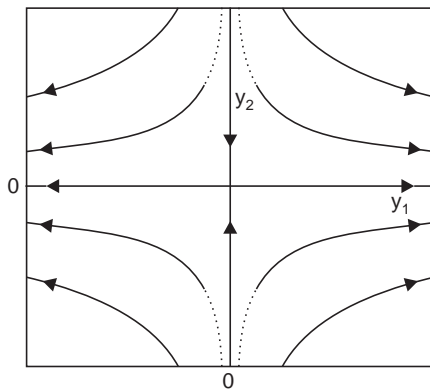



Fig. 14.31 The singular point is referred to as saddle point. $\lambda_2 < 0 < \lambda_1$ and $-\infty < \lambda_2/\lambda_1 < 0$

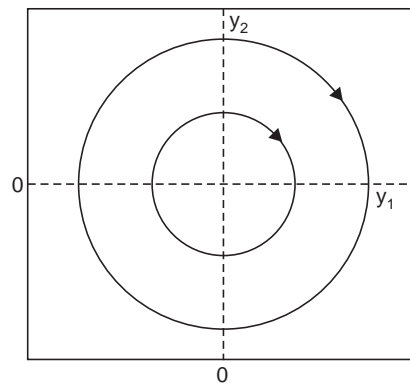



Fig. 14.32 The singular point is referred to as vortex. $\lambda_1 = \lambda_2 = \pm j\omega$, $\omega > 0$

(*d*) Case of a pair of complex conjugate roots with zero real part

The state portrait of the system equation (14.51) with complex conjugate roots shows unique characteristics. We show the phase portrait in Fig. 14.32 for the case with pure imaginary roots giving rise to a vortex or center.

(*e*) Case of a pair of complex conjugate roots with negative real part

The phase plane portrait in this case is a stable focus as shown in Fig. 14.33.

(*f*) Case of a pair of complex conjugate roots with positive real part

When the real part of the complex conjugate roots is positive, the phase portrait is an unstable focus as shown in Fig. 14.34.

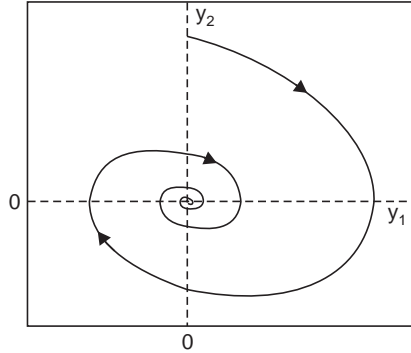



Fig. 14.33 The singular point is referred to as Stable focus $\lambda_{1,2} = -\sigma \pm j\omega \;\; \sigma > 0$

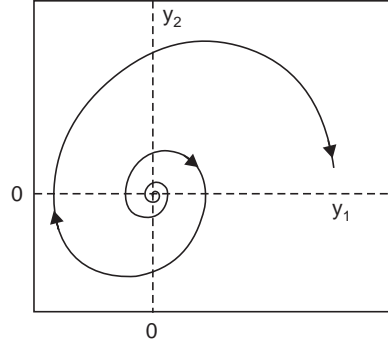



Fig. 14.34 The singular point is reffered to as unstable focus $\lambda_{1,2} = \sigma \pm j\omega \;\; \sigma > 0$

14.13 THE AIZERMAN AND KALMAN CONJECTURES

In connection with the stability of a class of nonlinear systems with the structure shown in Fig. 14.35(*a*), Aizermann put forth a conjecture [19], which may be stated as follows:

(*a*) **Aizermann's Conjecture:** If a single-loop system shown in Fig. 14.35(*b*) with a plant transfer function G(*s*) is such that the resultant closed-loop system 1 + KG(*s*) is stable for all linear gains K in the range $[k_1 \; k_2]$, then the nonlinear system shown in Fig. 14.35(*a*) incorporating the same linear plant transfer function and a memory-less nonlinearity $f(e)$ in sector $[k_1 \; k_2]$, is also stable where k_1, k_2 are defined by the relation

$$k_1 e^2 \le f(e)\, e \le k_2 e^2 \tag{14.52}$$

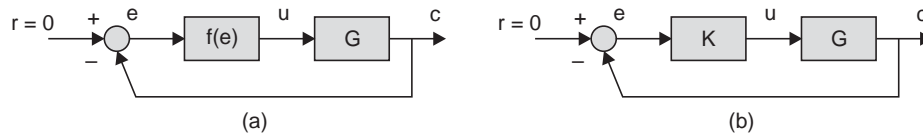



Fig. 14.35 (*a*) A single loop feedback system incorporating (*a*) nonlinear gain f(e) (*b*) equivalent linear gain K

The constants k_1, k_2 in relation (14.52) represent a linear envelope surrounding the nonlinearity vide Fig. 14.36. Even though Aizerman's conjecture appears to be reasonable, it has been shown to be false by counter-examples.

Kalman [19]suggested that the system of Fig. 14.35(*a*) will be stable provided that the linear system of Fig. 14.35(*b*) is stable for all *k* in the interval $[k_{1d}, k_{2d}]$ where

$$k_{1d} \le \frac{df(e)}{de} \le k_{2d} \;, \tag{14.53}$$

together with $\quad k_1 e^2 \le f(e) \le k_2 e^2 \quad$ and $\quad k_{1d} \le k_1 \le k_2 \le k_{d2} \qquad (14.54)$

Kalman's conjecture imposed additional conditions on the nonlinear characteristics but nevertheless it is also false—again shown by counter-examples. The failure of the two conjectures reveals that intuitive reasoning cannot be relied on in the case of nonlinear systems. One possible reason for the failure of the conjectures may be due to the presence of harmonics in nonlinear systems, which are absent in linear systems.

Now, we discuss below two techniques for dealing with the problem of stability of the class of systems shown in Fig. 14.35(*a*). These two techniques are (*a*) **Popov's Technique** and

the (*b*) **Circle Criteria**, which can be thought of as extensions of Nyquist's stability criterion for linear systems. These two techniques are presented below without proof.

14.13.1 Popov's Stability Criterion

Popov developed a graphical Nyquist-like criterion to examine the stability of the system shown in Fig. 14.35(*a*). It is assumed that open loop transfer function G(*s*) is stable. The nonlinearity *f*(*e*) must be time-invariant and piecewise continuous function of its input *e*. The derivative *df*(*e*)/*de* is bounded and *f*(*e*) satisfies the condition

$$0 \le f(e)\, e \le ke^2 \tag{14.55}$$

for some positive constant *k*. Graphically, the last condition means that the curve representing *f* must lie within a particular linear envelope. A *sufficient* condition *for global* asymptotic stability of the feedback loop may then be stated as:

If there exists any real number q and an arbitrarily small number $\delta > 0$ *such that*

$$\textit{Real part of } [(1 + j\omega q)\, G(j\omega)] + \frac{1}{k} \ge \delta > 0. \tag{14.56}$$

for all ω *then for any initial state the system output tends to zero as* $t \to \infty$.

It is convenient to define a modified transfer function G*(*j*ω) as shown below for carrying out a graphical test based on the above equation:

$$G^*(j\omega) = \text{Re}[G(j\omega)] + j\omega\text{Im}\,\{G(j\omega)\} \equiv X(j\omega) + jY(j\omega) \tag{14.57}$$

Therefore, in terms of X and Y the Popov's criterion becomes

$$X(j\omega) - qY(j\omega) + \frac{1}{k} \ge \delta > 0. \tag{14.58}$$

The so called Popov locus G*(*j*ω) is plotted in the complex plane. The system is then stable if some straight line, at an arbitrary slope 1/*q*, and passing through the –1/*k* point can be drawn that does not intersect the G*(*j*ω) locus (vide Fig. 14.37).

Since the Popov test gives only sufficient condition, failure to satisfy the criteria by a system does not imply that the system is unstable.

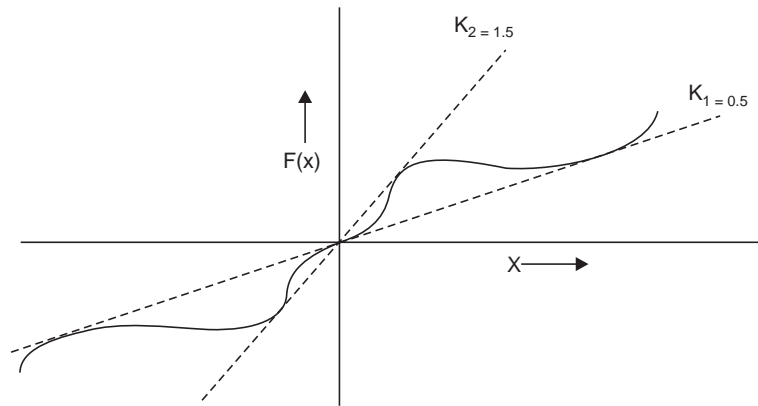



Fig. 14.36 Sector [k_2, k_1] containing the continuous valued function

14.13.2 The Generalized Circle Criteria

The circle criteria method of stability analysis can be considered as a generalization of Popov's method. The circle criteria method has two important advantages over the Popov's Method:

(*i*) It allows G(*s*) to be open loop unstable

(*ii*) It allows the nonlinearity to be time varying

The nonlinearity $f(e, t)$ is assumed to lie within an envelope such that,

$$k_1 e^2 < f(e, t)e < k_2 e^2, \tag{14.59}$$

similar to that shown in Fig. 14.36.

It is a sufficient condition for asymptotic stability of the system in Fig. 14.35(a) that the Nyquist plot G(jω) lies outside a circle in the complex plane that crosses the real axis at the points $-1/k_1$ *and* $-1/k_2$*, and has its center at the point,*

$$\frac{1}{2}\left[-\left(\frac{1}{k_1}+\frac{1}{k_2}\right)+j\omega q\left(\frac{1}{k_1}-\frac{1}{k_2}\right)\right] \tag{14.60}$$

for some real value of q.

In the above statement it is assumed that $k_1 < k_2$. The center of the circle is found to depend on both frequency and the value of q. Since the method gives a sufficient condition, the constant q can be set equal to zero and then a single frequency invariant circle results.

The circle can be considered as the generalization of the $(-1, j0)$ point in the Nyquist test for linear systems.

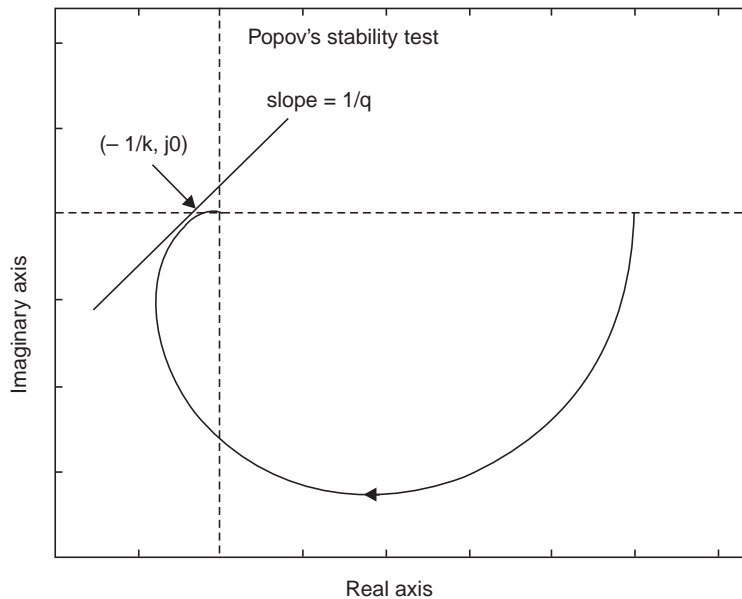



Fig. 14.37 The G*(*j*ω) locus and Popov's stability test

14.13.3 Simplified Circle Criteria

When the plant transfer function G(*s*) is minimum phase and open loop stable, the generalized circle criteria may be restated as:

The nonlinear system shown in Fig. 14.35(a) is stable if f(e) lies in sector $[k_1\ \ k_2]$*, where* $0 < k_1 < k_2$ *and the Nyquist plot of the linear plant transfer G(jω) does not intersect or encircle the disk (or circle) which is centered on the real axis and passes through the points* $(-1/k_1, j0)$ *and* $(-1/k_2, j0)$*.*

These conditions are *sufficient* but not *necessary* because intersection of the transfer function G(ω) and the disk as defined does not prove instability. The foregoing statements for the circle criteria is also valid for systems with feedback transfer function H(*s*), where the plant transfer function G(*s*) is to be replaced by loop gain G(*s*)H(*s*). The circle criteria is also applicable for the class of systems having structure shown in Fig. 14.35(*a*) where G(*s*) is replaced with discrete transfer function G(*z*).

14.13.4 Finding Sectors for Typical Nonlinearities

We shall consider nonlinear function $f(e)$ with a scalar input and a scalar output to define the sectors $[k_1, k_2]$. The nonlinearity $f(e)$ is said to belong to the sector $[k_1, k_2]$ if for all inputs e, the condition in relation (14.52) is satisfied.

In this definition k_1 or k_2 are allowed to assume values of $-\infty$ or $+\infty$. It is to be noted that k_1 and k_2 *do not* necessarily represent the minimum and maximum *slopes* of the function $f(e)$. In majority of cases k_1 and k_2 are, respectively, smaller than the minimum and maximum slopes. We illustrate below the computations of k_1 and k_2 for some typical cases.

Example 14.4 Sector for continuous function

With reference to the nonlinearity in Fig. 14.36, we draw two lines such that the relation (14.52) is satisfied. For the nonlinearity under reference, we find $k_1 = 0.5$ and $k_2 = 1.5$. It is to be found from the graph that the minimum of $\frac{df(e)}{de} = 0$ and maximum of $\frac{df(e)}{de}$ approaches ∞. The minimum and maximum of the slopes of the function definitely differs from k_1 and k_2 respectively.

Example 14.5 Sector for a relay with dead zone

A relay with dead zone is shown in Fig. 14.38. Since $f(e) = 0$ for $x \le d$, k_1 in relation (14.52) will be zero and $k_2 = M/b$. So the sector for the nonlinearity under reference is found as [0, 2], where M = 2 and $d = 1$. In this case k_1 is identical with the minimum slope of the nonlinearity, but k_2 differs from the maximum slope (which is ∞).

Example 14.6 Sector for an ideal relay (or sign function)

The characteristic for an ideal relay or sign function is shown in Fig. 14.39. In this case, since $f(e) = 0$ for $x = 0$, $k_1 = 0$ and $k_2 = M/0 = \infty$. The upper sector is coincident with the y-axis. So the sector is found as $[0, \infty]$. We shall get the same value for the sector for the sign function also. In this case, k_1 and k_2 are identical with the minimum and maximum slopes of the nonlinearity.

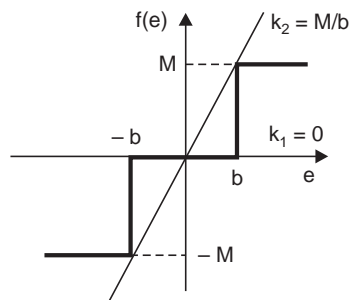



Fig. 14.38 Sector $[k_2, k_1]$ containing the relay with dead zone

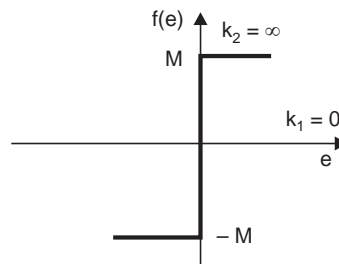



Fig. 14.39 Sector $[k_2, k_1]$ containing the ideal relay

Example14.7 Stability Based on the Circle Criterion

Find the sector $[k_1, k_2]$ for the type of nonlinearity shown in Fig. 14.36 by the simple circle criterion so that the system in Fig. 14.35(*a*) with G(*s*) given below is stable.

$$G(s) = \frac{25}{(s + 0.5)(s + 5)(s + 10)}$$

Solution: The relevant part of the Nyquist diagram of the transfer function $G(j\omega)$ is shown in Fig. 14.40. Since the circle criteria gives only sufficient condition of stability, the sector $[k_1, k_2]$ will not be unique. In Fig. 14.40, we have shown an arc of a circle in dash-dot line with radius approaching infinity and passing through the points $(-\infty, j0)$ and $(-0.083, j0)$ on the real axis. So the closed loop system will be stable for the class of nonlinearity contained in the sector [0, 12]. From the Nyquist diagrams of Fig. 14.40 we can also find solutions for sectors like [4.34, 17.24] and [11.23, 21.27] for stability, which shows that the sectors are not unique. It is to be noted that equivalent linear system is stable for $0 < K < 35.65$.

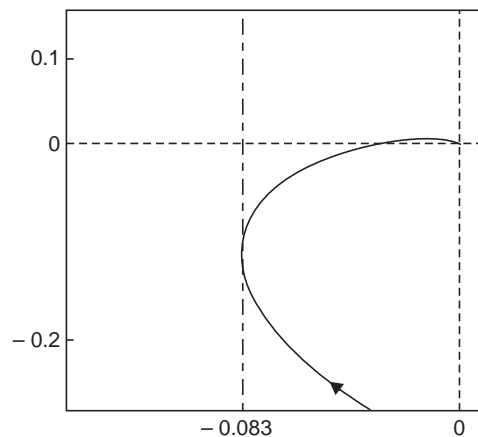



Fig. 14.40 Stability of Example 14.7 by circle criteria

We also note that stability of the closed loop system is assured for the above G(*s*) even if the nonlinearity is a relay with dead zone as shown in Fig. 14.38 with $M/b < 12$.

14.13.5 S-function Simulink solution of nonlinear equations

The S-function is a computer language description of dynamic systems, which uses special calling syntax that helps to interact with the equation solvers of **Simulink**. We shall consider here an illustrative example of writing M-file code for **user-defined S-function** for solving differential equations in MATLAB Version 6.5, Simulink version 5.0. We shall take the Van der Pol equation as an example and check its solution with other differential equation solvers available in Matlab.

We have, already, described the procedure of creating a new model in Simulink in Section 13.9 in Chapter 13.

Open **Simulink Library browser** *Simulink* in Matlab by following procedure of Section 13.9 and a **New Model.** Click **Continuous** to see the components of continuous system simulation, including integrator (1/*s*) blocks. With left mouse pressed drag two integrators blocks to the new model window, renamed integrator1 and integrator2 as shown in Fig. 14.41(*a*). For renaming a block, just click with left mouse and edit the block name. Similarly drag two blocks of **MuX** (found under node **Signal routing**), one block of **XY Graph** and one **(simout)**

To Workspace block (found under node **Sinks**) and a **(system) S-function** block found under **User-Defined Functions.**

Note that each block has one or more input ports and one output port. The **Mux** has more than one input ports. If required, more input ports may be added to the Mux block by selecting the Mux with right mouse-click and changing Mux parameters. Connect the blocks as shown in Fig. 14.41(*a*) by following the procedure in Section 13.9 of Chapter 13.

Now the state variable representation of Van der Pol equation $\ddot{x} - \varepsilon(1 - x^2)\dot{x} + x = 0$ is given by:

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = \varepsilon(1 - x_1^2)x_2 - x_1$$

We write the S-function with two inputs x_1 and x_2 and one **sys** output, which is $\dot{x}_2$(see Script_Fig14.1*a*). So, if the input to the first integrator is $\dot{x}_2$ then its output is x_2, which, by the state equation, is equal to $\dot{x}_1$. When $\dot{x}_1$ is the input to integrator2, its output will be x_1. The variables x_2 and x_1, which are respectively outputs from the first and second integrators, are used as feedback to the input of the **S-function** block through a Mux block. These variables are also connected to the **XY Graph** for the phase plane plot and can be sent to workspace by connecting them, through another **Mux**, to the block labelled **To Workspace**. So the diagram in Fig. 14.41*a* simulates the Van der Pol equation with the help of S-function realization. In order to run the simulation with given initial conditions, open **integrator1** by double clicking it and set desired initial condition, – 0.1, say and save it by clicking ok. Similarly, set initial condition for the **inegrator2** to 0.1. Open the S-function block and change the S-function name from system to Vander_Pol and save it. Open **simout** block by double clicking it and rename it as **var** (save format **array**), which stands for the variables x_1 and x_2. In order to get the phase plane plots with two sets of initial condition, we can duplicate the simulation diagram in part (*a*) to that shown in part (*b*). The simulink automatically renames the integrators as integrator 3 and integrator 4 and S-function block to S_function1, var to var1 etc. Set the desired initial conditions to integrator 3 and integrator 4 and save the model as **yourname.mdl** in the folder **work.** In order to write the M-file code for the S-function we can adapt a template of the S-function **sfuntmpl.m** found under Matlab\Toolbox\Simulink\Blocks. Copy the **sfuntmpl.m** to the folder **work** under **Matlab** and read it carefully to understand the template so as to adapt it for the solution of a dynamic system. The M-file code for S-function for the Van der Pol equation is listed in Script_Fig14.1*a*. Note that the name of the M-function Vander_Pol is the same name used in the S-function block in the simulink model **yourname.mdl.** Save the script as vander_pol.m file in the folder **work.** Now open **yourname.mdl,** select **Simulation parameter [Config. parameters in MATLAB 7.0]** under dropdown menu **simulation** and set **stop time** to suitable value, say, 25 sec. Now click **start** under simulation and observe the phase plane plots on the two XY Graphs. The final phase plane plot on a single display window can be obtained by entering the line of code below in MATLAB command window after running the model **yourname.mdl** from Simulink environment. Type in the line to get the phase plot in Fig. 14.1*a*.

```
>>plot(var(:,1),var(:,2)), hold on, plot(varl(:,1),var1(:,2),'r') grid on
```

We have discussed the procedure for creation of a user-defined S-function, which will be very helpful in simulating the dynamics of nonlinear systems in conjunction with the Simulink. We have also shown above how the phase plane plot may be obtained using an S-function and Simulink. The phase plot can also be obtained by using solvers ode25 or ode45 by creating an M-function file for the nonlinear equation (see fn141*b* and Script_Fig14.1*b* below).

```
%Script_Fig14.1a
%  M-file for user-defined S-function
function [sys,x0,str,ts] = vander_pol(t,x,xs,flag)  % save script as vander_pol.m  in
%work
switch flag,
case 0
    [sys,x0,str,ts]=mdlInitializeSizes;
case 3
    sys=mdlOutputs(t,x,xs);
case {1,2,4,9}
   sys=[ ];
otherwise
    error(['unhandled flag=',num2str(flag)]);
end
function [sys,x0,str,ts] = mdlInitializeSizes()
sizes=simsizes;
sizes.NumContStates=0;
sizes.NumDiscStates=0;
sizes.NumOutputs=1;
sizes.NumInputs=2;
sizes.DirFeedthrough=1;
sizes.NumSampleTimes=1;
sys=simsizes(sizes);
str=[ ];
x0=[ ];
ts=[-1 0];
function sys=mdlOutputs(t,x,xs)
 epsilon=0.4;
x1=xs(1);
x2=xs(2);
x2d=epsilon*(-1+x1*x1)*x2-x1;                 % Equation for Figure 14.1a
%x2d=-epsilon*(2)*x2-x1;                      % Equation for Figure  14.1b
sys=x2d;
%After running the Simulink model yourname.mdl from Simulink environment,
%type in MATLAB command window the following line to plot in Figure 14.1a
 %>>plot(var(:,1),var(:,2)),hold on, plot(var1(:,1),var1(:,2),'r'),grid on
```

```
% fn141b.m
% M-file for phase plane plots of fig 14.1 using solver ode25 or ode45.
function dxdt=fn141b(t,x)      % save script as fn141b.m in work
 epsilon=0.4;
 dxdt=[x(2);-epsilon*(1-x(1)^2)*x(2)-x(1)];
% dxdt must be arranged as column vector
% Remove  the negative sign before epsilon and make
% appropriate changes in initial conditions and time interval
%for obtaining phase plane plot of Fig 14.1a
```

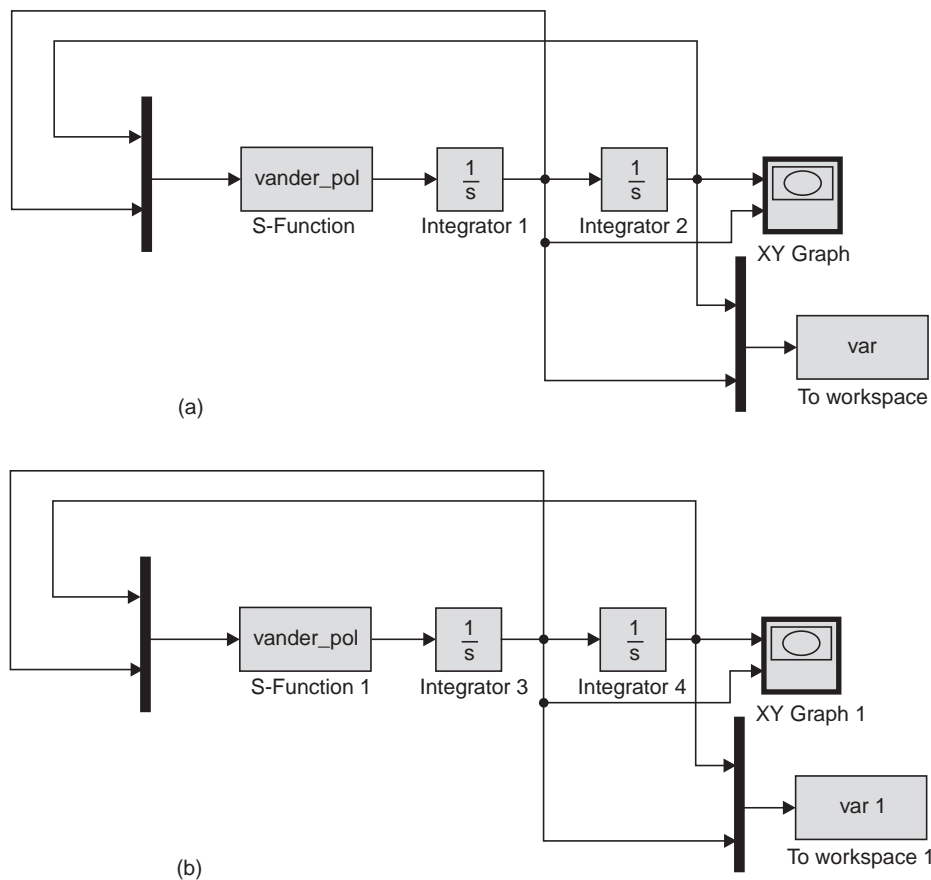



Fig. 14.41 Simulation diagram for obtaining phase plane plot for Van der Pol equation

```
%Script_Fig14.1b
% Phase plane plot for Van der Pol eqn d2xdt2=-0.epsilon(1-x^2)x-x
clear all; close all hidden
xi=[1;-0.5];
```

```
t1=0; t2=7.5;
[t,x]=ode45('fn141b',[t1  t2],xi);        % It integrates the differential
% equations dx/dt = f(t,x) from time t1 to t2 with initial conditions xi.
plot(x(:,1),x(:,2));           % phase plot in the x(1), x(2) plane
xlabel('x1\rightarrow'), ylabel('x2\rightarrow')
hold on
xxi=[1.2;1.35];               % another initial condition
[t,x]=ode45('fn141b',[t1  t2],xxi);
plot(x(:,1),x(:,2),'r'),;
% Make changes in the values of t1, t2 ,xi, xxi as indicated below and   %remove
negative sign before epsilon in the M-file vanderpol function
%to get the phase plane plot of Fig 14.1a
%xi=[0.1;-0.1]; t1=0; t2=25;  xxi = [2;3];
```

The M-file for the function **fn141b** and the Script_Fig14.1b should be saved in work as **fn141b.m** and **Fig14_1b.m** and type fig14_1b in the command window for execution

MATLAB SCRIPTS

```
%Script_Fig14.8
% Describing function of the nonlinearity in Figure 14.8a
%int(S, v, a, b) is the definite integral of S with respect to v
  syms k x t1 t2 t d b1 b2 y1 M
   y1=(sin(t)-sin(t1))*sin(t);
   y2= (sin(t2)-sin(t1))*sin(t)
   y3=(sin(t)-sin(t1))*sin(t); % y1,y2,y3 are integrands in
                        % equation (14.12a) with t = theta
    y4=int(y1,t1,t2)
    B11=2*y4;
    y5=int(y2,t2,pi-t2);
    B21=2*y5;
   y6=int(y3,pi-t2,pi-t1);
   B31=2*y6;
   B41=(B11+B21+B31)
  B1=simplify(B41) % Gives the result in equation (14.12b)
             %describing function N= |(A1/pi)+(B1/pi)|arctan (A1/B1)
```

```
 %Script_14.11
%Plot of Describing function of a relay with dead zone Refer Equ (14.19)
clear all; close all hidden
b=0.2; x=0.0:0.01:1.2; L=length(x);
 for j=1:L; x1=x/b;
```

```
warning off MATLAB:divideByZero
R=b/x(j);
if R >1
   B1=0;
else r1=1-power(R,2);
    r2=power(r1,0.5);
   B1=(4*r2)/(pi*x1(j));
end
NN(:,j)=B1;
end
  plot(x1,NN)
   axis([0 7 0 0.7]);
xlabel('X/b \rightarrow'), ylabel('N/M \rightarrow')
```

```
%Script_Fig14.17
% Amplitude and phase plot of the DF for backlash element
clear all; close all hidden
 syms k x t1 t2 t b1 b2 y1 M
   x=0.2:0.01:1.2; d=0.2;
x1=x/d; L=length(x);
 for j=1:L
 R=d/x(j);  A=-((4)*((1-R)*R))/(pi); %  refer to equ (14.26)
 AA(:,j)=A;
r1=(1/R)-1; r2=power(r1,0.5);
B=(1/pi)*((1-2*R)*2*R*r2+pi/2+asin(1-2*R));% Refer equ (14.25)
BB(:,j)=B;
N=B+i*A;
 N2=norm(N); % normalised DF
NN(:,j)=N2;
t1=(180/pi)*atan(A/B); % phase of DF in degrees
tt(:,j)=t1;
end
  subplot(2,1,1), plot(x1,NN)% Amplitude plot
 axis([0.5 6 0 1])
 xlabel('x/b \rightarrow'), ylabel('N/k \rightarrow')
subplot(2,1,2), plot(x1,tt)  % Phase plot
  axis([0.5 6 -95 0])
  xlabel('x/b \rightarrow'), ylabel('\leftarrow \phi')
```

```
%Script_Fig14.22
%Stability investigation of the system in Example 14.1 by DF method
clear all; close all hidden
x=0.04:0.01:0.2;
 d=0.00; b=0.04; M=10; x1=x/b;L=length(x);
for j=1:L
R1=d/x(j); R2=b/x(j); A=(2*M/(pi*x(j)))*(R1-R2);
r1=1-power(R1,2);r2=1-power(R2,2);
 B=(2*M/(pi*x(j)))*(power(r1,0.5)+power(r1,0.5));
 AA(:,j)=A; BB(:,j)=B;N=B+i*A;
 %N2=norm(N)/(2*M/b) ; % normalised DF
%NN(:,j)=N2;
%t1=(180/pi)*atan(A/B); % for obtaining phase plot
%tt(:,j)=t1;
N1=-1/N;B1=real(N1);B2(:,j)=B1;A1=imag(N1);A2(:,j)=A1;
end
plot(B2,A2)
 hold on
 g=zpk([],[0 -25 -5],32); %for  super imposing on Nyquist plot.
 nyquist(g)
```

```
% Script_fn1430a.m
%M-file for phase plane plots of fig 14.30a using solver ode25 or ode45.
function dxdt=fn1430a(t,x)
dxdt=[x(2);-7*x(2)-10*x(1)];
% dxdt must be arranged as column vector
```

```
%Script_Fig. 14.30
% Phase plane plot of a stable node d2x/dt2=-7dx/dt-10*x
clear all; close all hidden        % save script as Fig14_30.m in work
T=[0 10]; %Integration interval for the solver
xx=[-1 -0.5 0.5 1 1 0.5 -0.5 -1;2 2 2 2 -2 -2 -2 -2];  % set of initial conditions xi
for j=1:8; xi=xx(:,j);
[t,x]=ode45('fn1430a',T,xi);% It integrates the differential
% equations dx/dt = f(t,x) over the time interval T with initial conditions xi.
plot(x(:,1),x(:,2)); % phase plot in the x(1), x(2) plane
xlabel('x1\rightarrow'), ylabel('x2  \rightarrow')
hold on
end
```

REVIEW EXERCISE

RE 14.1 Find the describing function of an ideal relay having the characteristic shown in Fig. 14.39

The periodic output of the relay with sinusoidal input $x = X \sin \omega t = X \sin \theta$ is given by :

$$y(t) = \begin{cases} M, & 0 \le \theta \le \pi \\ -M, & \pi \le \theta \le 2\pi \end{cases}$$

Ans. $N = \dfrac{4M}{\pi X}$

RE 14.2 Find the describing function of an amplifier with static characteristic shown in Fig. RE 14.2

Hints. $y(t) = \begin{cases} 0 & , \; 0 \le \theta \le \theta_1 \\ kX \sin \theta - d, & \theta_1 \le \theta \le \pi - \theta_1 \\ 0 & , \; \pi - \theta_1 \le \theta \le \pi \end{cases}$

where $\theta_1 = \sin^{-1}(R)$, and $d/X = R$

Ans. $N = \dfrac{2k}{\pi}\left[\dfrac{\pi}{2} - \sin^{-1}R - R\sqrt{1 - R^2}\right]$

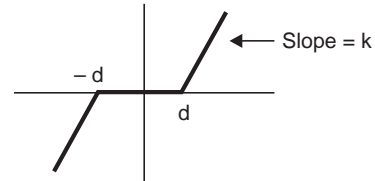



Fig. RE 14.2

PROBLEMS

P14. 1 Draw the phase plane plot of the system in Equation (14.44) by Delta Method.

P14.2 For the position control system with plant transfer function G(*s*) given by : $G(s) = \dfrac{5}{s(s+1)}$ incorporating a backlash element, show that the closed loop system with unity feedback will exhibit stable limit cycle oscillation. Compute the describing function N of the backlash element usnig relation (14.25) and (14.26) when $d = 1$ $k = 1$ and find the amplitude and frequency of oscillation following procedure of Section 14.6.

Ans. Frequency of oscillation $\omega = 1.6$ rad/sec and amplitude of oscillation , X = 2

P14.3 The open loop plant transfer function G(*s*) of a unity feedback system is given by: $G(s) = \dfrac{5e^{-0.1s}}{4s+1}$.

The controller preceding the plant is a relay with pure hysteresis [see Fig. 14.14(*a*)] having hysteresis width $b = 2$. Find the amplitude and frequency of limit cycle oscillation of the closed loop system by describing function method.

P14.4 The closed loop system with structure shown in Fig. 14.18, the nonlinearity is an ideal relay with characteristic given in RE14.1, where M = 10. Show, by describing function analysis, that the closed loop system incorporating a plant with transfer function G(*s*) given by: $G(s) = \dfrac{10}{s(s+10)}$ converges to the origin but oscillates with infinite frequency and zero amplitude.

P14.5 Repeat the analysis of problem P14.4 for a different plant with transfer function G(*s*) given by $G(s) = \dfrac{10000}{s(s+10)(s+40)}$ and show that the closed loop system exhibits stable limit cycle oscillations with frequency $\omega = 20$ rad/sec and amplitude 6.38.

P14.6 Investigate the stability by describing function method of a closed loop system with structure shown in Fig. 14.18, the nonlinearity being a relay with dead zone (but without hysteresis) as shown in Fig. 14.38 and the plant transfer function given by :

$$G(s) = \frac{720}{s(s+10)(s+40)}$$

The dead zone of the relay is $d = 0.5$ mA and M = 10.

P14.7 Investigate the stability of the problem P14.6 by circle criteria method.

CHAPTER

15

Implementation of Digital Controllers on Finite Bit Computer

15.1 INTRODUCTION

Digital controllers are implemented either in digital hardware or software on a microcomputer. In either case their bit size introduces numerous sources of error, which affects controller performance. The error depends on many factors including the way an algorithm is implemented to a program, the number system used to implement the control laws and the mathematical operations like truncation and rounding employed in the implementation. In this chapter we shall make an analysis of the various sources of error that affects controller performance. This will be useful to provide a guideline for the implementation of controller algorithm, choice of bit size of CPU, A/D and D/A converters, Memory word size, number system, operations like truncation and rounding so as to meet closed loop system performance.

15.2 IMPLEMENTATION OF CONTROLLER ALGORITHM

The control laws of a discrete system may be expressed either in state variable form or in transfer function form in z-domain. The state variable representation may be directly programmed on the microcomputer whereas the transfer function may be realized in two different ways: (*a*) using digital hardware (adders, summers, delay elements) and (*b*) programming on a computer.

15.2.1 Realization of Transfer Function

In both hardware and software realization of transfer function, there are number of alternative ways of doing it. We shall consider here the following four popular forms of realization :

(1) Direct form 1

(2) Direct form 2

(3) Cascade and

(4) Parallel form

The term **Canonical** is sometimes used in the literature to a realization in which the number of delay elements is equal to n, the order of the system and at other times to any realization with a minimum number of adders, multipliers and delay elements. **Direct form 1** and **Direct form 2** is sometimes referred to as **Series** realization and **Canonical** realization respectively. The choice of any particular realization depends on programming and on the numerical accuracy required as will be evident from the subsequent discussions.

15.2.2 Series or Direct Form 1

The transfer function of a controller is expressed as a ratio of two polynomials :

$$\frac{u(z)}{e(z)} = D(z) = \frac{\sum_{j=0}^{n} a_j z^{-j}}{1 + \sum_{j=1}^{n} b_j z^{-j}} \tag{15.1}$$

Hardware realization of relation (15.1) is shown in Fig. 15.1.

For software realization we write the difference equation from the transfer function in relation (15.1) as shown below:

$$u(k) = \sum_{j=0}^{n} a_j e(k-j) - \sum_{j=1}^{n} b_j u(k-j) \tag{15.2}$$

With a choice of states indicated in Fig. 15.1(*b*), we can write the following state variable representation of direct form 1 realization:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ . \\ . \\ . \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} -b_1 & 1 & 0 \ldots 0 & 0 \\ -b_2 & 0 & 1 \ldots 0 & 0 \\ . & & \cdots & \\ . & & & . \\ -b_{n-1} & & \ldots 0 & 1 \\ -b_n & 0 & \ldots 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ . \\ x_n(k) \end{bmatrix} + \begin{bmatrix} a_1 - a_0 b_1 \\ a_2 - a_0 b_2 \\ . \\ . \\ . \\ a_n - a_0 b_n \end{bmatrix} e(k) \tag{15.3}$$

$$u(k) = [1\ 0\ .\ .\ .\ 0] \begin{bmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ . \\ x_n(k) \end{bmatrix} + a_0 e(k) \tag{15.4}$$

(a) Non-minimum number of delay elements

(b) Minimum number of delay elements

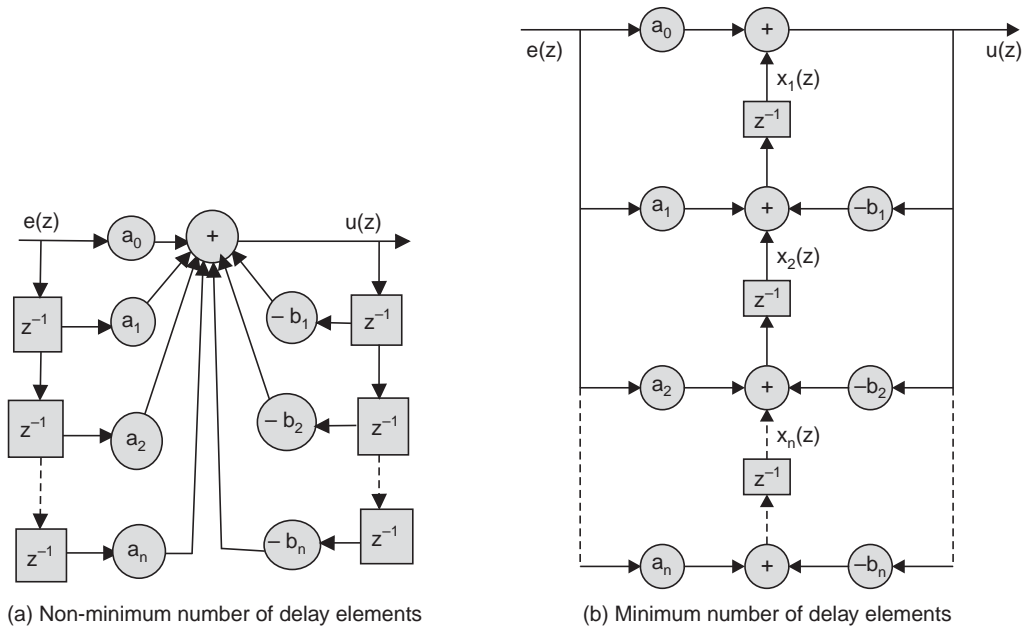



Fig. 15.1 Series or direct form 1 realization

15.2.3 Direct Form 2 (Canonical)

$$\frac{u(z)}{e(z)} = D(z) = \frac{\sum_{j=0}^{n} a_j z^{-j}}{1 + \sum_{j=1}^{n} b_j z^{-j}} = \frac{u(z)}{w(z)} \frac{w(z)}{e(z)}, \text{ where } w(z) \text{ is a dummy variable}$$

We can write the above relation as

$$\frac{w(z)}{e(z)} = \frac{1}{1 + \sum_{j=1}^{n} b_j z^{-j}} \tag{15.5}$$

and

$$\frac{u(z)}{w(z)} = \sum_{j=0}^{n} a_j z^{-j} \tag{15.6}$$

The hardware realization of Equation (15.5) and (15.6) is shown in Fig. 15.2.

Now, from relation (15.5) and (15.6) we can write, respectively, the difference equations needed for software realization as:

$$w(k) = e(k) - \sum_{j=1}^{n} b_j w(k-j) \tag{15.7}$$

$$u(k) = \sum_{j=0}^{n} a_j w(k-j) \tag{15.8}$$

With the choice of states as shown in Fig. 15.2(*b*), the state equation will be obtained as:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ . \\ . \\ . \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 & 0 \\ 0 & 0 & 1 & . & 0 & \\ . & & & . & & \\ . & & & . & & \\ . & & & 0 & 1 & \\ -b_n & -b_{n-1} & \ldots & -b_2 & -b_1 & \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ . \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ . \\ . \\ 0 \\ 1 \end{bmatrix} e(k) \tag{15.9}$$

and the controller output is found as:

$$u(k) = [(a_n - a_0 b_n) \ldots (a_1 - a_0 b_1)] \begin{bmatrix} x_1(k) \\ . \\ . \\ . \\ x_n(k) \end{bmatrix} + a_0 e(k) \tag{15.10}$$

(a) Non-minimum number of delay elements (b) Minimum number of delay elements

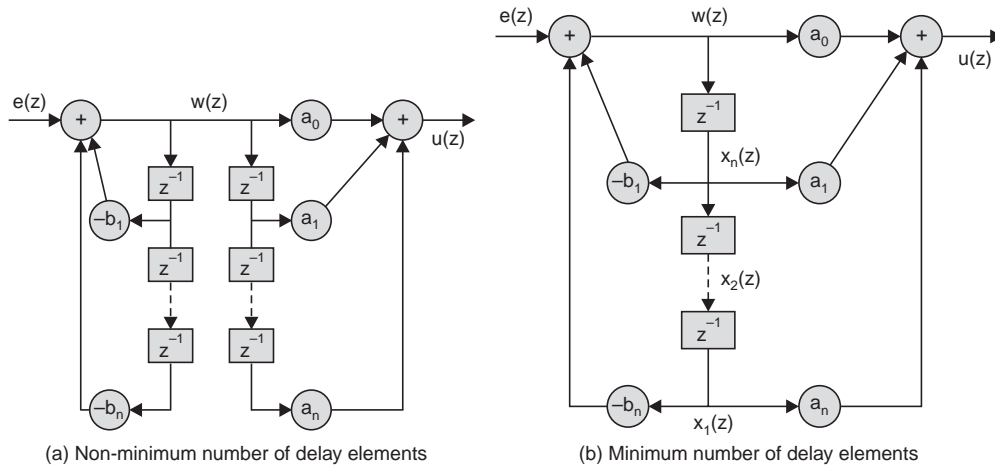



Fig. 15.2 Canonical or direct form 2 realization

15.2.4 Cascade Realization

The controller transfer function can be expressed in factored form when the poles and zeroes are known. The poles and zeros may be real or complex conjugate pairs. So, it can be written in terms of elementary blocks as :

$$\frac{u(z)}{e(z)} = D(z) = \alpha_0 D_1(z)\, D_2(z) \ldots D_m(z) \quad 1 \le m < n \tag{15.11}$$

where α_o is a DC term independent of frequency and the representative blocks $D_i(z)$ may consist of (*a*) a real zero and a real pole (*b*) a real zero and a complex pole pair and (*c*) complex pairs of zeros and poles. These blocks are shown below :

(*a*) Block of real zero and real pole:

$$D_1(z) = \frac{1+\alpha_1 z^{-1}}{1+\beta_1 z^{-1}} \tag{15.12}$$

(*b*) Block consisting of real zero and complex pole pair

$$D_m(z) = \frac{1+\alpha_m z^{-1}}{1+\beta_m z^{-1}+\beta_{m+1} z^{-2}} \tag{15.13}$$

where the coefficients in the numerator and denominator quadratic polynomials are real numbers.

(*c*) Block consisting of complex zeros and poles

$$D_m(z) = \frac{1+\alpha_m z^{-1}+\alpha_{m+1} z^{-2}}{1+\beta_m z^{-1}+\beta_{m+1} z^{-2}} \tag{15.14}$$

where all the coefficients in the numerator and denominator quadratic polynomials are real.

We note that, the transfer function in relation (15.12) can be obtained from relation (15.14) with substitution of $\alpha_m = \alpha_1$, $\alpha_{m+1} = 0$, $\beta_m = \beta_1$ and $\beta_{m+1} = 0$. So, we shall consider only the form in relation (15.14) in the following derivations, since the result corresponding to transfer function of (15.12) and (15.13) can be obtained with appropriate substitutions. for α's and β's.

Hardware realization: The hardware cascade realization for a controller transfer function $D_m(z)$, $m = 1$ and 2 given by relation (15.14) is depicted by the signal flow graph in Fig. 15.3 where z^{-1} stands for the delay element. For convenience of drawing signal flow graph, the dummy variables $E_1(z)$, $W_1(z)$ and $W_2(z)$ have been introduced such that the controller transfer function in (15.14) can be written as:

$$D_1(z) = \frac{1+\alpha_1 z^{-1}+\alpha_2 z^{-2}}{1+\beta_1 z^{-1}+\beta_2 z^{-2}} = \frac{W_2(z)}{W_1(z)} = \frac{E_1(z)}{W_1(z)} \cdot \frac{W_2(z)}{E_1(z)}$$

where $\frac{E_1(z)}{W_1(z)} = \frac{1}{1+\beta_1 z^{-1}+\beta_2 z^{-2}}$ and $\frac{W_2(z)}{E_1(z)} = 1+\alpha_1 z^{-1}+\alpha_2 z^{-2}$

Similarly, $D_2(z)$ is written as:

$$D_2(z) = \frac{1+\alpha_3 z^{-1}+\alpha_4 z^{-2}}{1+\beta_3 z^{-1}+\beta_4 z^{-2}} = \frac{U(z)}{W_2(z)} = \frac{E_2(z)}{W_2(z)} \cdot \frac{U(z)}{E_2(z)}$$

where $\frac{E_2(z)}{W_2(z)} = \frac{1}{1+\beta_3 z^{-1}+\beta_4 z^{-2}}$ and $\frac{U(z)}{E_2(z)} = 1+\alpha_3 z^{-1}+\alpha_4 z^{-2}$

Software realization: The software realization is achieved using the following difference state equations obtained with the choice of states indicated in Fig. 15.3.

$$x_1(k+1) = -\beta_1 x_1(k) - \beta_2 x_2(k) + \alpha_0 e(k)$$
$$x_2(k+1) = x_1(k)$$
$$x_3(k+1) = (\alpha_1 - \beta_1)\, x_1(k) + (\alpha_2 - \beta_2)\, x_2(k) - \beta_3 x_3(k) - \beta_4 x_4(k) + \alpha_0 e(k)$$
$$x_4(k+1) = x_3(k)$$
$$u(k) = (\alpha_1 - \beta_1)\, x_1(k) + (\alpha_2 - \beta_2)\, x_2(k) + (\alpha_3 - \beta_3)\, x_3(k) + (\alpha_4 - \beta_4)\, x_4(k) + \alpha_0 e(k)$$

So the state variable representation, in general, can be written as

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \\ x_{3(k+1)} \\ x_{4(k+1)} \\ . \\ . \\ . \\ x_{n(k+1)} \end{bmatrix} + \begin{bmatrix} -\beta_1 & -\beta_2 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ \alpha_1-\beta_1 & \alpha_2-\beta_2 & -\beta_3 & -\beta_4 & \ldots & 0 & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 & 0 \\ . & . & . & . & \ldots & . & . \\ . & . & . & . & \ldots & . & . \\ \alpha_1-\beta_1 & \alpha_2-\beta_2 & \alpha_3-\beta_3 & \alpha_4-\beta_4 & \ldots. & -\beta_{n-1} & -\beta_n \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ . \\ . \\ . \\ x_n \end{bmatrix} + \begin{bmatrix} \alpha_0 \\ 0 \\ \alpha_0 \\ 0 \\ . \\ . \\ . \\ \alpha_0 \end{bmatrix} e(k) \quad (15.15)$$

$$u(k) = [(\alpha_1 - \beta_1)\,(\alpha_2 - \beta_2)\ \ldots\ (\alpha_{n-1} - \beta_{n-1})\,(\alpha_n - \beta_n)] \begin{bmatrix} x_1(k) \\ x_2(k) \\ . \\ . \\ . \\ x_n(k) \end{bmatrix} + \alpha_0 e(k) \quad (15.16)$$

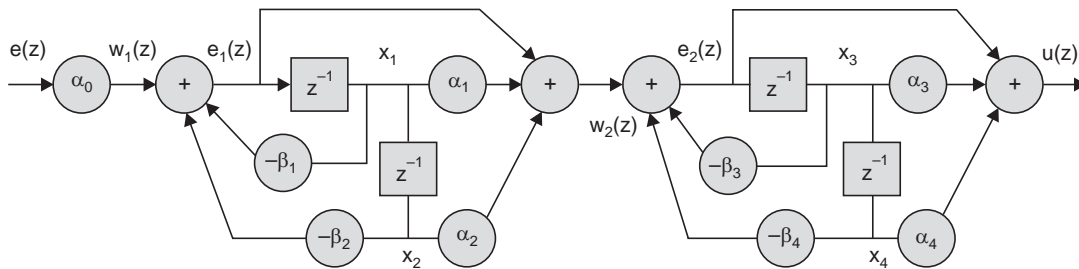



Fig.15.3 Cascade realization of controller

15.2.5 Parallel Realization

The controller transfer function can also be expressed by partial fraction expansion as shown below when its poles are known :

$$\frac{u(z)}{e(z)} = \gamma_0 + D_1(z) + D_2(z) + \ldots + D_m(z), \quad 1 < m < n \quad (15.17)$$

There can be two representative forms of $D_m(z)$ corresponding to simple and complex poles as shown below.

(*a*) The general form corresponding to real pole

$$D_1(z) = \frac{\gamma_1}{1 + \beta_1 z^{-1}} \quad (15.18)$$

(b) The general quadratic form corresponding to a complex conjugate pole pair

$$D_m(z) = \frac{\gamma_m + \gamma_{m+1}\, z^{-1}}{1+\beta_m z^{-1} + \beta_{m+1} z^{-2}} \tag{15.19}$$

We note, as before, that the transfer function of (15.18) can be obtained from the transfer function in (15.19) with appropriate substitutions of γ's and β's. So, in the following analysis we shall use the form in relation (15.19) and the results corresponding to the transfer function in (15.18) can be written with appropriate substitutions.

Hardware realization with two factors of the forms in (15.19), is shown in Fig. 15.4.

Software realization: The software realization is achieved using the discrete state representation obtained for the choice of states shown in Fig. 15.4.

$$x_1(k+1) = -\beta_1 x_1(k) - \beta_2 x_2(k) + e(k)$$
$$x_2(k+1) = x_1(k)$$
$$x_3(k+1) = -\beta_3 x_3(k) - \beta_4 x_4(k) + e(k)$$
$$x_4(k+1) = x_3(k)$$

and $$u(k) = (\gamma_2 - \gamma_1\beta_1)\, x_1(k) - \gamma_1\beta_2 x_2(k) + (\gamma_4 - \gamma_3\beta_3)\, x_3(k) - \gamma_3\beta_4 x_4(k) + (\gamma_0 + \gamma_1 + \gamma_3)\, e(k)$$

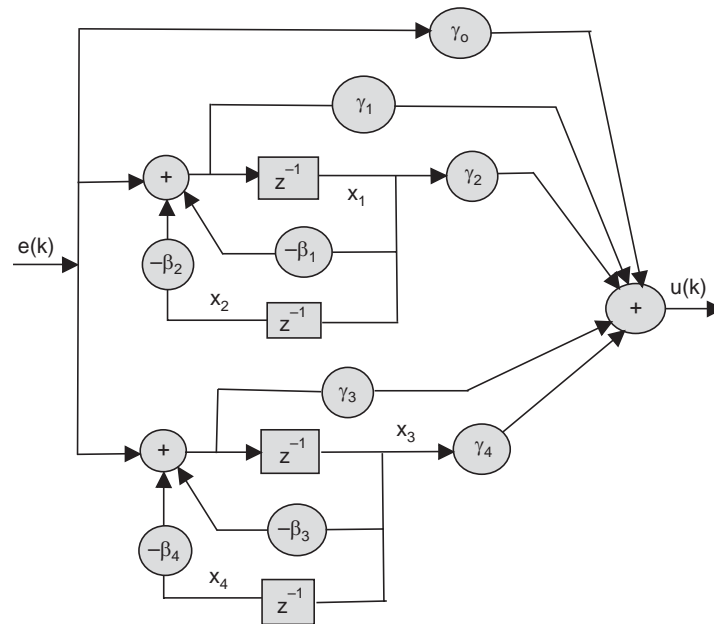



Fig. 15.4 Parallel realization of controller

The matrix representation for parallel realization, in general, is of the form :

$$\begin{bmatrix} x_{1(k+1)} \\ x_{2(k+1)} \\ x_{3(k+1)} \\ x_{4(k+1)} \\ \cdot \\ \cdot \\ \cdot \\ x_{n(k+1)} \end{bmatrix} = \begin{bmatrix} -\beta_1 & -\beta_2 & 0 & 0\ldots & 0 & 0 \\ 1 & 0 & 0 & 0\ldots & 0 & 0 \\ 0 & 0 & -\beta_3 & -\beta_4\ldots & 0 & 0 \\ 0 & 0 & 1 & 0\ldots & 0 & 0 \\ \cdot & \cdot & \cdot & \cdot\ldots & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot\ldots & \cdot & \cdot \\ 0 & 0 & 0 & 0\ldots & -B_{n-1} & -\beta_n \\ 0 & 0 & 0 & 0\ldots & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \\ \cdot \\ \cdot \\ \cdot \\ x_n(k) \end{bmatrix}_i + \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ 1 \\ 0 \end{bmatrix} e(k) \tag{15.20}$$

$$u(k) = [(\gamma_2 - \gamma_1\beta_1) - \gamma_1\beta_2 \ldots (\gamma_n - \gamma_{n-1}\beta_{n-1}) - \gamma_{n-1}\beta_n] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix}$$

$$+ (\gamma_o + \gamma_1 + \gamma_3 + \ldots + \gamma_{n-1})\, e(k) \qquad (15.21)$$

Example 15.1 We now illustrate the various realizations with a numerical example.

Let $\dfrac{u(z)}{e(z)} = \mathrm{D}(z) = \dfrac{2 + 0.6z^{-1} - 0.2z^{-2}}{1 - 0.65z^{-1} + 0.1z^{-2}}$

(*i*) Series form (Direct I)

$$u(k) = 2e(k) + 0.6e(k-1) - 0.2e(k-2) + 0.65u(k-1) - 0.1u(k-2)$$

or
$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.65 & 1 \\ -0.1 & 0 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1.9 \\ -0.4 \end{bmatrix} e(k)$$

$$u(k) = [1 \quad 0] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 2e(k)$$

(*ii*) Direct form 2

$$w(k) = e(k) + 0.65w(k-1) - 0.1\, w(k-2)$$

$$u(k) = 2e(k) + 1.9w(k-1) - 0.4\, w(k-2)$$

or,
$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.1 & 0.65 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}_i + \begin{bmatrix} 0 \\ 1 \end{bmatrix} e(k)$$

$$u(k) = [-0.4 \quad 1.90] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 2e(k)$$

(*iii*) Cascade

$$\frac{u(z)}{e(z)} = \mathrm{D}(z) = \frac{2(z+0.5)\,(z-0.2)}{(z-0.4)\,(z-0.25)} = \frac{2(1+0.5z^{-1})\,(1-0.2z^{-1})}{(1-0.4z^{-1})\,(1-0.25z^{-1})}$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.4 & 0 \\ 0.9 & 0.25 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}_i + \begin{bmatrix} 2 \\ 2 \end{bmatrix} e(k)$$

$$u(k) = [0.9 \quad 0.05] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 2e(k)$$

(*iv*) Parallel

$$\frac{u(z)}{e(z)} = \frac{2(z+0.5)\,(z-0.2)}{(z-0.4)\,(z-0.25)} = 2 + \frac{2.4z^{-1}}{1-0.4z^{-1}} - \frac{0.5z^{-1}}{1-0.25z^{-1}}$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0.4 & 0 \\ 0 & 0.25 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} e(k)$$

$$u(k) = [2.4 - 0.5] \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + 2e(k)$$

The direct method is the easiest to implement but the method generates the largest numerical errors (vide Section 15.6). The cascade and the parallel realization generate smaller errors for the same word length. Therefore, realization in direct form should not be used except for a very low order controller.

15.3 EFFECTS OF FINITE BIT SIZE ON DIGITAL CONTROLLER IMPLEMENTATION

The implementation of digital controller on a microcomputer with finite bit size introduces a number of sources of errors. The error is dependent on the number system used to implement the control laws and the mathematical operations like truncation and rounding. We shall make an analysis of the nature and character of the error by considering two different number systems.

15.3.1 Sign Magnitude Number System (SMNS)

Let us assume that x is a real number and it is normalized i.e., $0 \leq | x | \leq 1$, and $Q(x)$ is the Quantized version of x in a finite number of bits.

15.3.1.1 Truncation Quantizer

In a microcomputer with b number of bits to represent amplitude of the quantized number, the truncation operation retains only b number of bits after the binary point and the rest are discarded. So, there will be an error between the magnitude of the analog signal or coefficient and the quantized value represented with finite bits. This error e, may be defined as:

$$e = Q_t(x) - x,$$

where $Q_t(x)$ is the quantized number and t stands for truncation operation. Using s as the sign bit (0 for positive, 1 for negative), the quantized number with b number of bits with truncation operation can be written as :

$$Q_t^b(x) = (s.m_1 \; m_2 \; m_3 \; . \; . \; .m_b)_{2smns}$$

Now, the number x with unlimited number of bits is given by :

$$x = (s.m_1 \; m_2 \; m_3 \; . \; . \; ..m_b) + 2^{-b} (s.\, m_{b+1} \; m_{b+2} \; . \; . \; .)_{2smns}$$
$$= Q_t^b + 2^{-b} (s.\, m_{b+1} \; m_{b+2} \; . \; . \; .)_{2smns}$$

Case 1. x is positive, For $x \geq 0$, $| x | \geq Q_t^b(x)$ and

$$e_t = Q_t^b(x) - x = | Q_t^b(x) | - | x | = -(.0\;0 \; . \; . \; m_{b+1}\, m_{b+2} \; . \; . \; .)_{2smns}$$
$$= -2^{-b} (0.\, m_{b+1}\, m_{b+2} \; . \; . \; .)_{2smns}$$

But $0 \leq (0.\, m_{b+1} \; m_{b+2} \; . \; . \; .)_{2smns} < 1$

Therefore $\mathbf{0 \geq e_t > -2^{-b}}$, for $\mathbf{x \geq 0}$ (15.22)

Case 2. x is negative For the case $x < 0$,

$$e_t = Q_t^b(x) - x = -\{| Q_t^b(x) | - | x |\}$$

so that $\mathbf{0 \leq e_t < 2^{-b}}$, for $x < 0$ (15.23)

The quantization characteristics and the error is plotted in Fig. 15.5 for SMNS with truncation operation.

15.3.1.2 Round-Off Quantizer

In rounding off quantizer, if b number of bits are used to represent the magnitude of the quantized number, rounding operation is performed with $b + 1$ number of bits, then only b number of bits counted from the binary point, are retained. One bit is used to represent the sign of the number. As before we consider the case of positive and negative number separately.

Case 1: x is positive, $x \geq 0$

$$x = (s\,.\,n_1 \; n_2 \ldots n_b \; n_{b+1} \; n_{b+2} \ldots)_{2smns}$$
$$+ 2^{-b-1} = (0\,.\,0 \;\; 0 \ldots 0 \;\; 1 \;\;\; 0 \ldots)_{2smns}$$

$$x + 2^{-b-1} = \{s.\, m_1\, m_2 \ldots m_b\, (n_{b+1} \oplus 1)\, n_{b+2} \ldots\}_{2smns}$$
$$= Q_r^b(x) + 2^{-b}(0.\,(n_{b+1} \oplus 1)\, n_{b+2} \ldots)_{2smns}$$

where $\oplus$ stands for exclusive or operation and $Q_r^b(x) = (s.\ m_1\ m_2 \ldots\ldots\ldots m_b)_{2smns}$ (assuming no overflow).

or $$Q_r^b(x) - x - 2^{-b-1} = -2^{-b}\,(0.(n_{b+1} \oplus 1)\ n_{b+2} \ldots.)_{2smns}$$

$$\rightarrow \quad e_r - 2^{-b-1} = -2^{-b}\,(0.(n_{b+1} \oplus 1)\ n_{b+2} \ldots)_{2smns}$$

where the error is defined as $e_r = Q_r^b(x) - x$.

$\rightarrow$ $0 \geq e_r - 2^{-b-1} > -2^{-b}$, since $0 \leq (0.\ (n_{b+1} \oplus 1)\ n_{b+2} ..) < 1$

$\rightarrow$ $0 \geq e_r - 2^{-b}/2 > -2^{-b}$. Now adding $2^{-b}/2$ we get

$2^{-b}/2 \geq e_r > -2^{-b} + 2^{-b}/2$; or $\boldsymbol{q/2 \geq e_r > -q/2}$, **for** $\boldsymbol{x \geq 0}$, (15.24)

where $\boldsymbol{q = 2^{-b}}$ is the least significant bit.

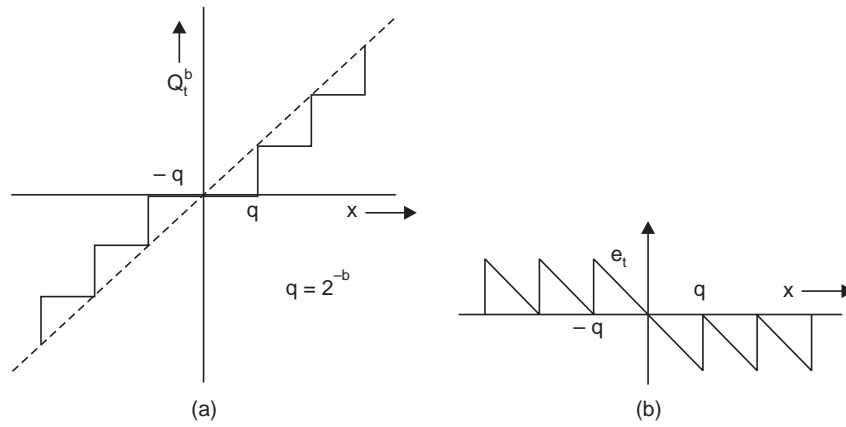



Fig. 15.5 Quantizer characteristic for SMNS with truncation operation (*a*) quantizer characteristic (*b*) error plot

Case 2: *x* is negative, $x < 0$

$$\begin{aligned} x &= (1\ .\ n_1\ n_2\ \ldots\ n_b\ n_{b+1}\ n_{b+2} \ldots\ldots)_{2smns} \\ +\,2^{-b-1} &= (0\ .\ 0\ \ 0\ \ldots\ 0\ \ 1\ \ 0\ \ldots\ldots)_{2smns} \\ \hline x + 2^{-b-1} &= \{1\ .\ m_1\ m_2 \ldots\ldots\ m_b\ (n_{b+1} \oplus 1)\ n_{b+2} ..\}_{2smns} \\ &= Q_r^b(x) + 2^{-b}\,(0.\ (n_{b+1} \oplus 1)\ n_{b+2} ..)_{2smns} \end{aligned}$$

where $Q_r^b(x) = (1.\ m_1\ m_2 \ldots\ldots\ldots\ m_b)_{2smns}$

or $$Q_r^b(x) - x - 2^{-b-1} = -2^{-b}\,(0.\ (n_{b+1} \oplus 1)\ n_{b+2} \ldots.)_{2smns}$$

$\rightarrow$ $e_r - 2^{-b-1} = -2^{-b}\,(0.\ (n_{b+1} \oplus 1)\ n_{b+2} \ldots)_{2smns}$ where $e_r = Q_r^b(x) - x$.

$\rightarrow$ $0 \geq e_r - 2^{-b-1} > -2^{-b}$, since $0 \leq (0.\ (n_{b+1} \oplus 1)\ n_{b+2} \ldots) < 1$

$\rightarrow$ $0 \geq e_r - 2^{-b}/2 > -2^{-b}$. Now adding $2^{-b}/2$ we get

$2^{-b}/2 \geq e_r > -2^{-b} + 2^{-b}/2$; or $\boldsymbol{q/2 \geq e_r > -q/2}$ **for** $x < 0$, **where** $\boldsymbol{q = 2^{-b}}$. (15.25)

The quantization characteristics and the error is plotted in Fig. 15.6 for SMNS with rounding operation. The error density function for truncation as well as rounding operation are depicted in Fig. 15.7 which is a continuous function of error *e*. This is due to the fact that *x* is a real number and equally likely to assume positive and negative values, hence the error *e* is continuous. The height of density function will be such as to make the area under the distribution curve equal to unity.

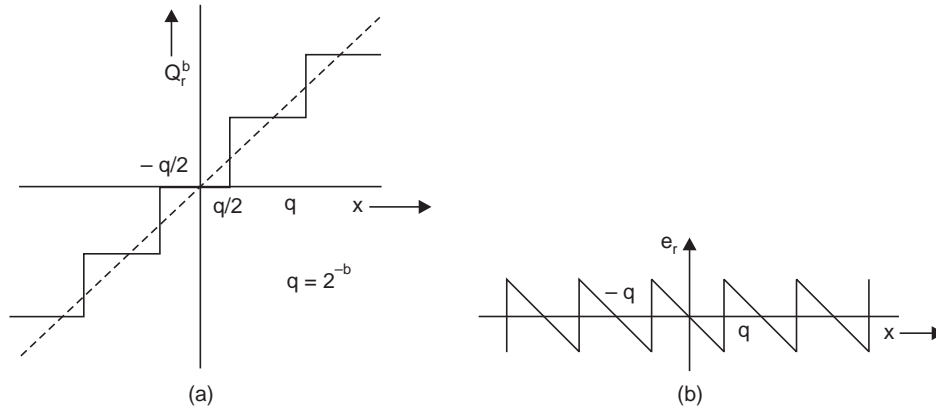



Fig. 15.6 Quantizer Characteristic for SMNS with Rounding operation (*a*) quantizer characteristic (*b*) error plot

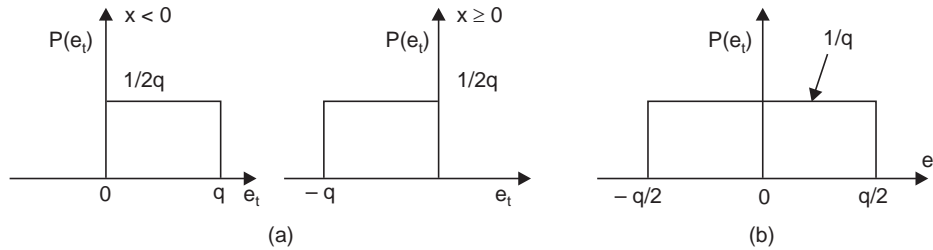



Fig. 15.7 Quantification error probability density function for the signed magnitude number system (*a*) Truncation (*b*) Round -off.

15.3.1.3 Mean and Variance

We have to use the statistical parameters for a measure of errors introduced with finite bit representation of numbers in a computer with truncation and rounding operation. The parameters of our interest are mean and variance and we shall employ the following relation for computing the variance of the discretization error e after truncation and rounding operation.

The variance of the error, $\sigma_e^2 = E(e^2) - E^2(e)$ (15.26)

where $E(e)$ stands for expected value of e and $E^2(e) = \{E(e)\}^2$

A. Truncation operation with SMNS

Now for truncation operation the probability of e_t is shown in Fig. 15.7(*a*). Therefore we have

$$\sigma_{et}^2 = \{E(e_t^2) - E^2(e_t)\}_{x \geq 0} + \{E(e_t^2) - E^2(e_t)\}_{x < 0} \tag{15.27}$$

Now for positive error, the expected value of error is found as :

$$E[e_t]_{x \geq 0} = \int_{-q}^{0} p(e_t) e_t \, de_t = \int_{-q}^{0} \frac{1}{2q} e_t \, de_t = \frac{1}{2q} \frac{e_t^2}{2} \Big|_{-q}^{0} = -\frac{q}{4}$$

and,

$$E[e_t^2]_{x \geq 0} = \int_{-q}^{0} e_t^2 \cdot \frac{1}{2q} de_t = \frac{1}{2q} \frac{e_t^3}{3} \Big|_{-q}^{0} = \frac{q^2}{6}$$

Similarly for negative error, the expected value is : $E[e_t]_{x < 0} = \frac{q}{4}$ and $E[e_t^2]_{x < 0} = \frac{q^2}{6}$

Therefore, using relation (15.27) and simplification we get $\sigma_{et}^2 = \dfrac{5q^2}{24}$ (15.28)

And the mean is given by : $E(e_t) = -\dfrac{q}{4} + \dfrac{q}{4} = 0$ (15.29)

B. Rounding operation with SMNS

Using the error probability density function for rounding operation of Fig. 15.7(*b*), and proceeding as above the variance and mean can be computed as :

Variance, $\boldsymbol{\sigma_{er}^2 = q^2/12}$ and mean, $\boldsymbol{E(e_r) = 0}$ (15.30)

So, the variance with rounding operation is found to be reduced by a factor of 2.5 compared to that of the truncation operation. Therefore, the rounding operation will produce $\sqrt{2.5} = 1.58$ times more accurate result over truncation operation with the same number of bits.

15.3.1.4 Dynamic Range of SMNS

The dynamic range (D.R.) of number system for microcomputer with b number of bits gives an idea about the maximum permissible level of controller output. It is defined as:

$$\text{D.R.} = \frac{\text{Largest magnitude } (|Q^b(x)|_{\max})}{\text{smallest non-zero magnitude } (|Q^b(x)|_{\min} \neq 0} = (1 - 2^{-b})/2^{-b} = 2^b - 1$$

Therefore, the numbers must fall in the range $\boldsymbol{-(2^{-b} - 1) \le 2^b . Q^b(x) \le 2^b - 1}$ (15.31)

15.3.1.5 Overflow

Now, suppose we add two numbers in the binary SMNS.

$$\begin{array}{l} Q^b(x_1) = (s_1 . m_{11}\, m_{12} \ldots m_{1b})_{2smns} \\ Q^b(x_2) = (s_2 . m_{21}\, m_{22} \ldots m_{2b})_{2smns} \\ \hline Q^b(x_3) = (s_3 . m_{31}\, m_{32} \ldots m_{3b})_{2smns} \end{array}$$

If s_1 and s_2 are different, then

$$s_3 = s_1, \text{ if } |Q^b(x_1)| \ge |Q^b(x_2)|;\ s_3 = s_2, \text{ if } |Q^b(x_1)| < |Q^b(x_2)|$$

and $$|Q^b(x_3)| = \big|\, |Q^b(x_1)| - |Q^b(x_2)| \,\big|$$

But if s_1 and s_2 are the same then; $s_3 = s_1 = s_2$; and $|Q^b(x_3)| = |Q^b(x_1)| + |Q^b(x_3)|$ and $|Q^b(x_3)|$ may exceed the range of the number system.

If the addition is accomplished using a binary adder and the overflow bit is ignored, the resulting overflow characteristic for the sign magnitude number system is shown Fig. 15.8. It is apparent that the plant will be subjected to wrong controller output in case of overflow. The analogue controller output may be more tolerable under similar situation *i.e.* when it saturates.

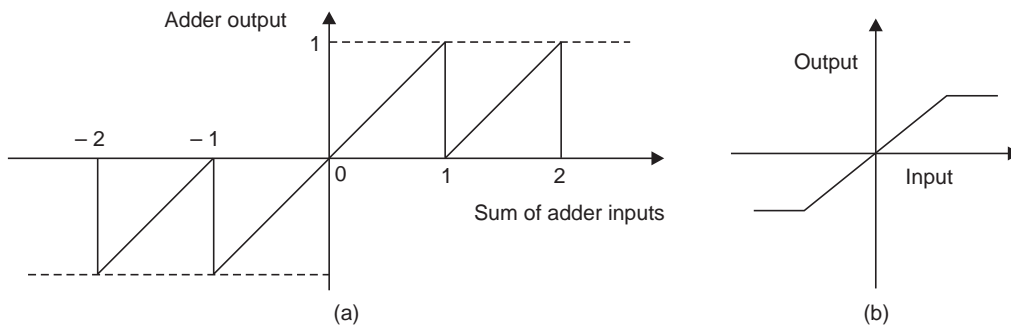



Fig. 15.8 (*a*) Overflow characteristics in SMNS (*b*) Saturation in analogue controller

15.3.2 Two's Complement Number System

15.3.2.1 Truncation Operation

$$Q^b(x) = (0.\ m_1\ m_2\ \ldots\ m_b)_{2cns} \quad \text{for} \quad 0 \le x < 1$$
$$= (1.\ n_1\ n_2\ \ldots\ n_b)_{2cns} \quad \text{for} \quad -1 \le x < 0$$

Case 1. x is positive. For positive numbers we may use the results of the SMNS analysis

$$\mathbf{0 \ge e_t > -2^{-b},\ x \ge 0}$$

Case 2. x is negative, $x < 0$

$$x = (1.\ n_1\ \ n_2\ \ldots\ n_b\ n_{b+1}\ \ldots)_{2cns}$$
$$= Q_t^{\ b}(x) + 2^{-b}(0.\ n_{b+1}\ n_{b+2}\ \ldots.)_{2cns}$$
$$Q_t^{\ b}(x) - x = e_t = -2^{-b}(0.\ n_{b+1}\ n_{b+2}\ \ldots)_{2cns}$$

Therefore, $\mathbf{0 \ge e_t > -2^{-b}}$ **for $x < 0$**

Hence for all x, $\mathbf{0 \ge e_t > -q,\ q = 2^{-b}}$ (15.32)

The quantization characteristics is shown in Fig. 15.9 (a). The probability density function is shown in Fig. 15.9(b).

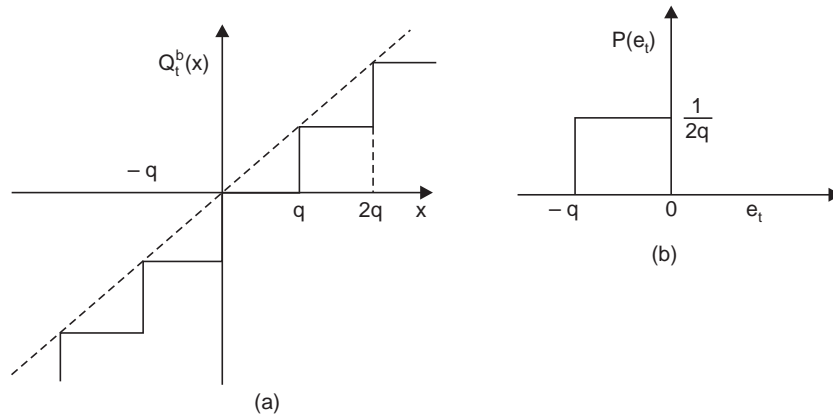



Fig. 15.9 Quantization characteristics in Two's CNS with truncation operation (a) quantization characteristics (b) probability density function

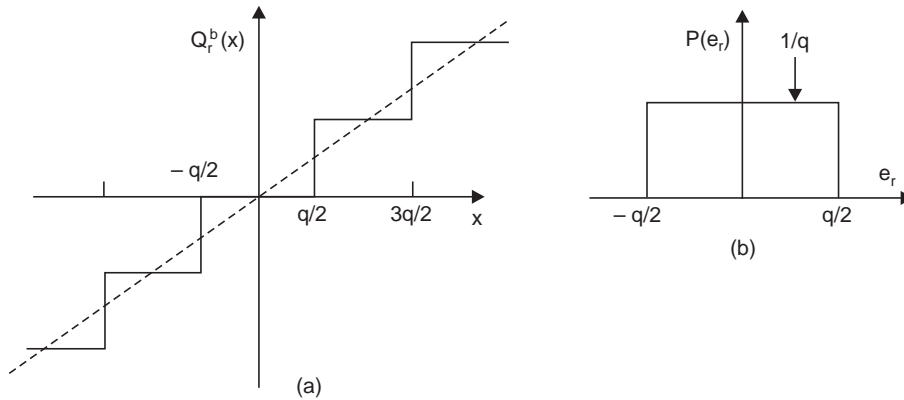



Fig. 15.10 Quantization characteristics in Two's CNS with rounding operation (a) quantization characteristics (b) probability density function

15.3.2.2 Round-Off Quantizer in Two's CNS

Case 1. *x* is positive,

Now suppose that we consider the case of the round-off quantizer in determining $Q_r^{\,b}(x)$. For positive numbers, round-off in the two's complement number system duplicates the sign magnitude case.

$$\text{since } 0 \le \sum_{i=1}^{\infty} n_{b+i} 2^{-b} < 1$$

Hence $\quad \boldsymbol{q/2 \ge e_r > -q/2, \quad x \ge 0}$ (15.33)

Case 2. *x* is negative

$$\begin{array}{rl} x = & (1\,.\; n_1\; n_2 \ldots n_b\; n_{b+1}\; n_{b+2}\; \ldots)_{2\text{cns}} \\ +\,2^{-b-1} = & (0\,.\; 0\; 0\; \ldots\; 0\; 1\; 0\; \ldots)_{2\text{cns}} \\ \hline x + 2^{-b-1} = & (1\,.\; m_1\; m_2 \ldots m_b\; (n_{b+1}\oplus 1) n_{b+2} \ldots)_{2\text{cns}} \\ = & Q_r^{\,b}(x) + 2^{-b}[0.(n_{b+1}\oplus 1)\; n_{b+2}\ldots]_{2cns} \end{array}$$

or $\quad Q_r^{\,b}(x) - x - 2^{-b-1} = -2^{-b}[0.(n_{b+1}\oplus 1)\; n_{b+2} \ldots]_{2cns}$

$\rightarrow \quad 0 \ge e_r - 2^{-b-1} > -2^{-b}$; Now adding $2^{-b}/2$ we get

$\rightarrow \quad 2^{-b}/2 \ge e_r > -2^{-b} + 2^{-b}/2$

$\rightarrow \quad 2^{-b}/2 \ge e_r > -2^{-b}/2$

that is $\mathbf{q/2 \ge e_r > -q/2}$, for $\mathbf{x < 0}$ (15.34)

Consequently, round-off error bound for the two's CNS is the same as that for SMNS. The quantizer characteristics is shown in Fig. 15.10(*a*) and the error distribution is shown in Fig. 15.10(*b*).

15.3.2.3 Mean and Variance

A. Truncation operation with two's CNS

The mean and variance of 2's CNS with truncation are calculated as follows:

The error probability density for this case is shown in Fig. 15.9(*b*). Therefore, we have

$$E[e_t] = \int_{-q}^{0} e_t\, p(e_t)\, de_t = \int_{-q}^{0} \frac{1}{q} e_t de_t = \frac{1}{q}\frac{e_t^2}{2}\Big|_q^0 = -\frac{q}{2}$$

and
$$E[e_t^2] = \int_{-q}^{0} e_t^2 \cdot \frac{1}{q} de_t = \frac{1}{q}\frac{e_t^3}{3}\Big|_{-q}^0 = \frac{q^2}{3}$$

Hence,
$$\sigma_{et}^2 = E[e_t^2] - E^2[e_t] = \frac{q^2}{3} - \left(-\frac{q}{2}\right)^2 = \frac{q^2}{3} - \frac{q^2}{4} = \frac{q^2}{12}$$

and the mean $= \dfrac{-q}{2}$

The variance of truncation operation with Two's CNS is found to be the same as that of the rounding quantizer of SMNS.

The mean value for the truncation operation is found to be nonzero and it introduces a negative dc value, which will cause an offset in the controller output and its effect on the closed loop system should be carefully studied.

B. Rounding Operation With Two's CNS

Considering the error density function of Fig. 15.10(b) and using the Equation (15.26), we can find out the variance and the mean of the quantization noise for the Two's CNS with rounding operation as :

$$\sigma_{er}^2 = \frac{q^2}{12} \tag{15.35}$$

and $$E[e_r] = 0$$

15.3.2.4 Dynamic Range for Two's CNS

Since $-1 \leq Q^b(x) \leq 1 - 2^{-b}$, the dynamic range for Two's complement number system is given by:

$$(\text{D.R.})_{2cns} = \{\left| Q^b(x) \right|_{\max}\}/\{\left| Q^b(x) \right|_{\min} \neq 0\} = \left| -1 \right| / \left| 2^{-b} \right| = 2^b$$

15.3.2.5 Overflow

When two numbers are added to generate a third, the binary sum along with the signs are automatically taken care of by the hardware adder circuits. When, two negative numbers are added, the carry bit into the 2^1 position is ignored by the hardware. So, when two negative numbers are added and a carry bit into sign position becomes zero the adder output will swing from most negative to most positive value. Similarly, when two positive numbers are added and a carry bit into sign position becomes one the adder output will jump from most positive to most negative value. Hence the overflow characteristic for two's complement number system will be as shown in Fig. 15.11.

Suppose that in a closed loop control system, the error signal is computed by the microcomputer, which is converted to analog signal by a D/A converter and applied to the dc motor after proper amplification. In case of an overflow in the error channel, the dc motor will be subjected to instant reversal of the direction of rotation, which is a dangerous situation, apart from the controller's failure for error correction.

$$Q^b(x_1) = (m_{10} \,.\, m_{11} \;\; m_{12} \,\ldots\ldots\ldots m_{1b})_{2cns}$$
$$Q^b(x_2) = (m_{20} \,.\, m_{21} \;\; m_{22} \,\ldots\ldots\ldots m_{2b})_{2cns}$$
$$\overline{Q^b(x_3) = (m_{30} \,.\, m_{31} \;\; m_{32} \,\ldots\ldots\ldots m_{3b})_{2cns}}$$

Sign bit (pointing to m_{30})

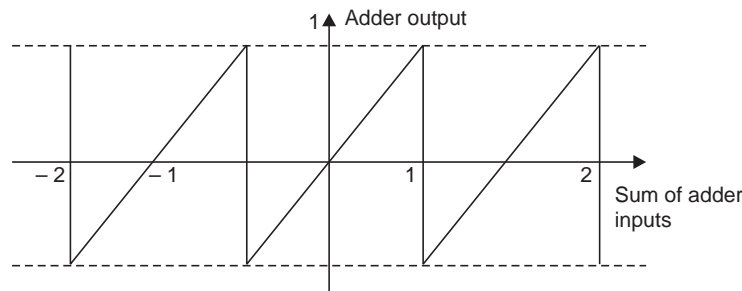



Fig. 15.11 The overflow characteristics in two's complement number system

15.4 PROPAGATION OF QUANTIZATION NOISE THROUGH THE CONTROL SYSTEM

In the previous section, we have found the bound of noise generated by truncation and round-off quantizer in SMNS and Two's CNS. The quantized noise acts as additional input and propagates through the controller. We are interested to find out the effect of the noise on the controller output when implemented with different implementation schemes discussed in Section 15.2.

It will be found that the propagation of the quantization noise, e, depends on the transfer function between the source point of error $\boldsymbol{e}$ and the controller output. It is assumed that the system is linear and time invariant and is characterized by its transfer function $D(z)$ and that the input noise e is characterized by its mean $\bar{e}$ and its variance σ_e^2. Since the mean value of noise, $\bar{e}$ is constant, $e(z) = \dfrac{z}{z-1}\bar{e} = \dfrac{1}{1-z^{-1}}\bar{e}$. So, the controller output may be written as $u(z) = \dfrac{\bar{e}}{1-z^{-1}} D(z)$ and by final value theorem, we can write

$$\bar{u}(\infty) = \bar{u} = \lim_{z \to 1} (1 - z^{-1}) \frac{\bar{e}}{1-z^{-1}} D(z) = \bar{e}\, D(1)$$

If, for example, $D(z) = \dfrac{z}{z-\beta}$, $\bar{u} = \bar{e}\, D(1) = \dfrac{\bar{e}}{1-\beta}$

The variance of the output signal u is calculated by Perseval theorem from a given pulse transfer function. If $\dfrac{u(z)}{e(z)} = D(z)$, the variance σ_u^2 at the output is given by the line integral around $|z| = 1$

$$\sigma_u^2 = \sigma_e^2 \frac{1}{2\pi j} \oint D_k(z) \,.\, D_k(z^{-1})\, z^{-1} dz \tag{15.36}$$

The line integral can be computed by evaluating the residues of $D_k(z)\, D_k(z^{-1}) z^{-1}$ at the poles of $D_k(z)$.

Example15.2 $\dfrac{u(z)}{e(z)} = D(z) = \dfrac{1}{1-\beta_1 z^{-1}} = \dfrac{z}{z-\beta_1}$

Therefore, $D(z^{-1}) = \dfrac{1}{1 - z\beta_1}$; so the residue R of $D_k(z)\, D_k(z^{-1})\, z^{-1}$ at the poles of $D(z)$ is given by

$$R = \lim_{z \to \beta_1} (z - \beta_1) \frac{z}{(z-\beta_1)(1 - z\beta_1)} z^{-1} = \frac{1}{1-\beta_1^2}$$

Therefore, $$\sigma_u^2 = \sigma_e^2 \frac{1}{2\pi j} \oint D(z).D(z^{-1})\, z^{-1} dz = \sigma_e^2 \frac{1}{1-\beta_1^2}$$

15.5 VERY HIGH SAMPLING FREQUENCY INCREASES NOISE

The choice of sampling frequency for digital control system is very important. It is normally taken as 10 times the bandwidth of system transfer function. However, if sampling frequency is chosen at a high value it increases the digitization noise. This is illustrated by the following example.

Consider an analog controller with transfer function D(*s*) given by

$$D(s) = \frac{u(s)}{e(s)} = \frac{1}{s+\alpha}$$

such that $D(z) = \frac{z}{z - e^{-\alpha T}} = \frac{z}{z-\beta}$, where $\beta = e^{-\alpha T}$ and T is the sampling period.

The variance of the noise at the controller output is given by $\sigma_u^2 = \sigma_e^2 \frac{1}{1-\beta^2}$; where $\sigma_e^2 = \frac{q^2}{12}$, $q = 2^{-b}$ when b is the number of bits used to represent the digital number.

In Table 15.1, the variation of $\frac{1}{1-\beta^2}$ is presented as function of the sampling period T with $\alpha = 2$

Table 15.1 Noise variance at the controller output with sampling frequency

T	β	$\frac{1}{1-\beta^2}$
0.100	0.8187	3.0332
0.050	0.9048	5.5167
0.010	0.9802	25.5033
0.001	0.9980	250.5005

From Table 15.1 it is found that the variance of noise will increase rapidly with the increase of sampling frequency. The obvious explanation is that due to the fast sampling rate the difference between two successive numbers is small and only the least significant bits are changing.

15.6 PROPAGATION OF ADC ERRORS AND MULTIPLICATION ERRORS THROUGH THE CONTROLLER

A controller algorithm may be realized by a number of alternative structures of which direct form 1; direct form 2, cascade and parallel realizations have been discussed in Section 15.2. The analysis of the estimated noise that will be generated for a particular transfer function using a given structure is not readily available. We shall, rather, demonstrate the phenomenon using an example to bring out some basic rules for the guidance of design engineer.

Consider a controller transfer function given by

$$\frac{u(z)}{e(z)} = D(z) = \frac{\gamma_1}{1-\beta_1 z^{-1}} + \frac{\gamma_2}{1-\beta_2 z^{-1}} \tag{15.37}$$

15.6.1 Propagated Multiplication Noise in Parallel Realization

We will investigate the parallel realization first to examine how the scaling influences the error propagation. Next, we shall compare the error propagation between parallel realization and a direct realization.

Two parallel realizations of the controller transfer function in relation (15.37) are shown in Fig. 15.12. The analog signal *e* is converted by a finite bit A/D converter to get the digitized

signal e_1. In Fig. 15.12(*a*) when e_1 is multiplied with the constant γ_1 stored in a memory location of finite bit *b* and retain only *b* number of bits for the product after the operation of truncation or rounding, an error is generated represented by ε_1. The error ε_1 is treated as an additional input to the controller as shown in the Fig. 15.12(*a*). Similarly ε_2, ε_3, ε_4 accounts for the errors generated from the product of the relevant signals with β_1, γ_2 and β_2 respectively. Similarly, we show the error components in Fig. 15.12(*b*).

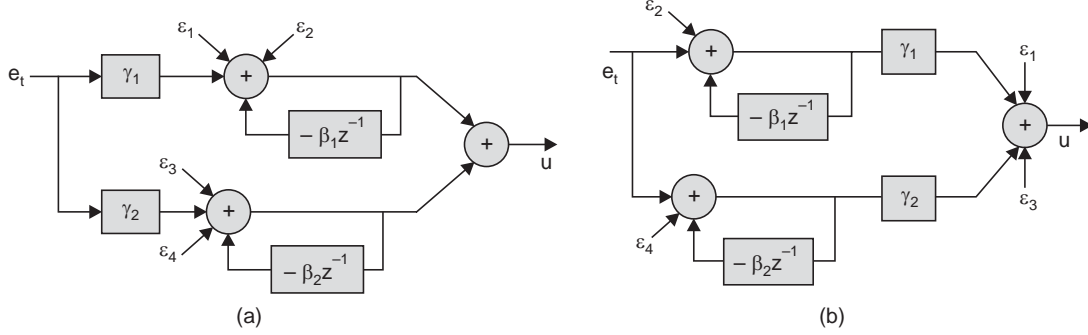



Fig. 15.12 Parallel realization of the controller in equation (15.37)

We are interested to study the effect of the propagated noise at the controller output. It is apparent that the propagation of the noise depends on the transfer function between the source point of the error and the controller output. The propagation of the noise to the controller output will be studied by computing the statistical parameters of the output noise when the statistical parameters like mean and variance of the input noise is known. The transfer functions of the various noise inputs can be found out from Fig. 15.12. With reference to Fig. 15.12(*a*), the transfer functions to noise inputs are given by :

$$\frac{u(z)}{\varepsilon_1(z)} = D_1(z) = \frac{1}{1-\beta_1 z^{-1}} = \frac{z}{z-\beta_1}, \qquad \frac{u(z)}{\varepsilon_2(z)} = D_2(z) = \frac{1}{1-\beta_1 z^{-1}} = \frac{z}{z-\beta_1}$$

$$\frac{u(z)}{\varepsilon_3(z)} = D_3(z) = \frac{1}{1-\beta_2 z^{-1}} = \frac{z}{z-\beta_2}, \qquad \frac{u(z)}{\varepsilon_4(z)} = D_4(z) = \frac{1}{1-\beta_2 z^{-1}} = \frac{z}{z-\beta_2}$$

Therefore, $\bar{u} = 2\bar{\varepsilon}_m \left\{\frac{1}{1-\beta_1} + \frac{1}{1-\beta_2}\right\}$, where it is assumed that $\varepsilon_m = \varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4$

The variance is computed by finding the residues at β_1 and β_2

The residue evaluated at $z = \beta_1$ is given by

$$R_1 = (z-\beta_1)\,\frac{z}{(z-\beta_1)}\,\frac{z^{-1}}{(z^{-1}-\beta_1)}\,z^{-1}\Big|_{z=\beta_1} = \frac{1}{1-\beta_1^2}$$

Therefore, $\sigma_u^2 = 2\sigma_\varepsilon^2 \left\{\frac{1}{1-\beta_1^2} + \frac{1}{1-\beta_2^2}\right\}$, where it is assumed that $\sigma_\varepsilon^2 = \sigma_{\varepsilon_1}^2 \ldots = \sigma_{\varepsilon_4}^2$ (15.38*a*)

We can proceed in the same way to find the mean and variance of the propagated noise at the controller output for the implementation shown in Fig. 15.12(*b*). Under the same simplifying assumptions regarding the amplitudes of different noise components, these are given by :

$$\bar{u} = \bar{\varepsilon}_m \left\{\frac{\gamma_1}{1-\beta_1} + \frac{\gamma_2}{1-\beta_2} + 2\right\} \text{ and } \sigma_u^2 = \sigma_\varepsilon^2 \left\{\frac{\gamma_1^2}{1-\beta_1^2} + \frac{\gamma_2^2}{1-\beta_2^2} + 2\right\} \qquad (15.38b)$$

Obviously, if both γ_1 and $\gamma_2 > 1$, the noise is more pronounced in the realization shown in Fig. 15.12(b) compared to that of the realization in part (a) of the Figure.

15.6.2 Propagated Multiplication Noise in Direct form Realization

We shall now find the mean and variance of the propagated noise at the controller output, when the controller in Equation (15.37) is realized in a direct from. We rearrange the controller transfer function as shown below and depict its simulation diagram in Fig. 15.13.

$$\frac{u(z)}{e(z)} = \frac{\gamma_1}{1-\beta_1 z^{-1}} + \frac{\gamma_2}{1-\beta_2 z^{-1}} = \frac{a_0 + a_1 z^{-1}}{1+b_1 z^{-1} + b_2 z^{-2}} \tag{15.38c}$$

where $a_0 = \gamma_1 + \gamma_2,\ a_1 = -(\gamma_1\beta_2 + \gamma_2\beta_1),\ b_1 = -(\beta_1 + \beta_2),\ b_2 = \beta_1\beta_2$

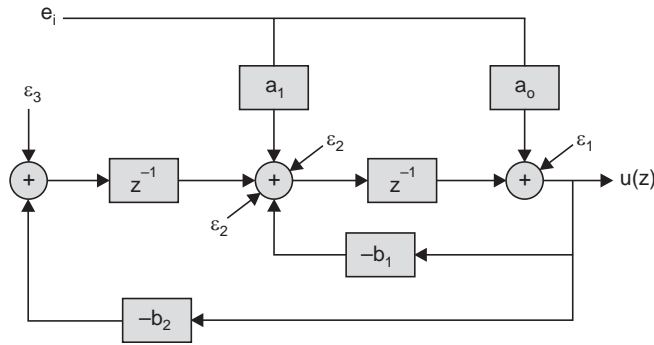



Fig. 15.13 Cascade realization of the controller in equation (15.37)

The noise components are also shown at appropriate points in the diagram. Proceeding as before, the corresponding transfer functions from the noise inputs to the controller output, the mean $\bar{u}$ and the noise variance are found as follows :

For the noise input ε_1

$$\frac{u_1(z)}{\varepsilon_1(z)} = \frac{1}{1+b_1 z^{-1} + b_2 z^{-2}} = \frac{z^2}{(z-\beta_1)(z-\beta_2)}$$

$$\bar{u}_1 = \bar{\varepsilon}_1 \frac{1}{(1-\beta_1)(1-\beta_2)}$$

$$\sigma_{u_1}^2 = \sigma_{\varepsilon_1}^2 \frac{1}{(\beta_1-\beta_2)(1-\beta_1\beta_2)}\left[\frac{\beta_1}{1-\beta_1^2} - \frac{\beta_2}{1-\beta_2^2}\right]$$

For the noise input ε_2

$$\frac{u_2(t)}{\varepsilon_2(t)} = \frac{z^{-1}}{1+b_1 z^{-1} + b_2 z^{-2}} = \frac{z}{(z-\beta_1)(z-\beta_2)}$$

$$\bar{u}_2 = \bar{\varepsilon}_2 \frac{1}{(1-\beta_1)(1-\beta_2)}$$

$$\sigma_{u_2}^2 = \sigma_{\varepsilon_2}^2 \frac{1}{(\beta_1-\beta_2)(1-\beta_1\beta_2)}\left[\frac{\beta_1}{1-\beta_1^2} - \frac{\beta_2}{1-\beta_2^2}\right]$$

For the noise input ε_3

$$\frac{u_3(z)}{\varepsilon_3(z)} = \frac{z^{-2}}{1+b_1 z^{-1} + b_2 z^{-2}} = \frac{1}{(z-\beta_1)(z-\beta_2)}$$

$$\bar{u}_3 = \bar{\varepsilon}_3 \frac{1}{(1-\beta_1)(1-\beta_2)}$$

$$\sigma_{u_3}^2 = \sigma_{\varepsilon_3}^2 \frac{1}{(\beta_1 - \beta_2)(1-\beta_1\beta_2)}\left[\frac{\beta_1}{1-\beta_1^2} - \frac{\beta_2}{1-\beta_2^2}\right] \quad (15.39)$$

The total mean of u caused by the function noises ε_1, $2\varepsilon_2$, ε_3 is given by

$$\bar{u}_m = \bar{u}_1 + 2\bar{u}_2 + \bar{u}_3 = \frac{4\bar{\varepsilon}_m}{(1-\beta_1)(1-\beta_2)}, \text{ assuming } \varepsilon_m = \varepsilon_1 = \varepsilon_2 = \varepsilon_3$$

and the total variance of u caused by ε_1, $2\varepsilon_2$ and ε_3 is given by

$$\sigma_u^2 = \sigma_\varepsilon^2 \frac{4}{(\beta_1 - \beta_2)(1-\beta_1\beta_2)}\left[\frac{\beta_1}{1-\beta_1^2} - \frac{\beta_2}{1-\beta_2^2}\right], \text{ assuming } \sigma_\varepsilon^2 = \sigma_{\varepsilon_1}^2 = \sigma_{\varepsilon_2}^2 = \sigma_{\varepsilon_3}^2 = \sigma_{\varepsilon_4}^2$$

$$= 4\,\sigma_\varepsilon^2 \frac{1+\beta_1\beta_2}{(1-\beta_1\beta_2)(1-\beta_1^2)(1-\beta_2^2)} \quad (15.40)$$

We may compare this noise amplification to the noise amplification in the parallel realization (vide Equation 15.38*a*)

$$\frac{\sigma_u^2 \text{ Parallel}}{\sigma_u^2 \text{ Direct}} = \frac{(2-\beta_1^2-\beta_2^2)(1-\beta_1\beta_2)}{2(1+\beta_1\beta_2)} \quad (15.41)$$

For high sample rates, the numerical values of β_1 and β_2 are very close to 1. Consequently, the ratio given by equation (15.41) shows that the parallel realization has lower multiplication noise amplification.

The results for comparison of the mean, propagated through the parallel and the direct realization can be shown to be given by

$$\frac{\bar{u} \text{ Parallel}}{\bar{u} \text{ Direct}} = \frac{(1-\beta_1)+(1-\beta_2)}{2} = \frac{2-\beta_1-\beta_2}{2} \quad (15.42)$$

15.7 COEFFICIENT ERRORS AND THEIR INFLUENCE ON CONTROLLER DYNAMICS

It is well known in analog design procedures that some particular implementation of a control network may be more sensitive to the variation of parameters than others. The same form of sensitivity is encountered in the implementation of the digital controller. Because of the finite word length, the coefficients of the controller D(z) are slightly different from their pre-calculated value. The difference causes variation of the pole and zero locations of the controller D(z). This sensitivity depends on the numerical structure of the controller D(z). It may be shown that some structure of realization are more sensitive than others.

15.7.1 Sensitivity of Variation of Coefficients of a Second Order Controller

Let us consider a controller transfer function given by:

$$D(z) = \frac{u(z)}{e(z)} = \frac{\gamma_1}{1-\beta_1 z^{-1}} + \frac{\gamma_2}{1-\beta_2 z^{-1}} \quad \text{(Parallel form)}$$

$$= \frac{(\gamma_1+\gamma_2) - (\gamma_1\beta_2 + \gamma_2\beta_1)z^{-1}}{1-(\beta_1+\beta_2)z^{-1} + \beta_1\beta_2 z^{-2}} \quad \text{(Direct form)}$$

$$= \frac{\alpha_1}{(1-\beta_1 z^{-1})} \cdot \frac{\alpha_2}{(1-\beta_2 z^{-1})} \quad \text{(Cascade form)}$$

(*a*) Direct form of realization

Characteristic polynomial in the Direct form is given by :

$$P(z) = z^2 - (\beta_1 + \beta_2)z + \beta_1\beta_2$$

$\rightarrow$ $\quad P(z) = z^2 - \lambda_1 z + \lambda_2$ where $\lambda_1 = \beta_1 + \beta_2$ and $\lambda_2 = \beta_1\beta_2$

In order to consider the sensitivity of pole locations to a change in coefficients, we shall let P(z, λ) be the characteristic equation, where the variation of coefficients λ_I's by $\delta\lambda_i$ yield relocation of poles to $z_1 + \delta z_1$, $z_2 + \delta z_2$

Therefore, we can write, $P(z_j + \delta z_j, \lambda + \delta\lambda) = P(z_j, \lambda) + \dfrac{\partial P}{\partial z_j}\delta z_j + \dfrac{\partial P}{\partial \lambda}.\delta\lambda$

By definition of characteristics equation, the left hand side and first term on the right hand side are zero.

Hence, $$\frac{\partial z_j}{\partial \lambda} = -\left.\frac{\partial P/\partial\lambda}{\partial P/\partial z}\right|_{z = z_j}$$

Considering a second order characteristic polynomial of the form, $P = z^2 - \lambda_1 z + \lambda_2$

we get, $$\frac{\partial P}{\partial \lambda_1} = -z, \; \frac{\partial P}{\partial z} = 2z - \lambda_1, \; \frac{\partial P}{\partial \lambda_2} = 1$$

Consequently, $$\frac{\partial z}{\partial \lambda_1} = -\left.\frac{-z}{2z - \lambda_1}\right|_{z = \beta_1} = \frac{\beta_1}{2\beta_1 - (\beta_1 + \beta_2)} = \frac{\beta_1}{\beta_1 - \beta_2}$$

and $$\frac{\delta z}{\delta \lambda_2} = \left.\frac{-1}{2z - \lambda_1}\right|_{z = \beta_2} = \frac{-1}{\beta_1 - \beta_2}$$

For closely related poles ($\beta_1 \cong \beta_2$), the direct realization, is found to be highly sensitive to variation in coefficients.

(*b*) The cascade and parallel realization

In cascade and parallel realization, the coefficients, mechanized in the algorithm are the poles themselves.

So, the characteristic equation : $P = z - \beta_1$. Therefore, $\dfrac{\partial P}{\partial z} = 1$, $\dfrac{\partial P}{\partial \beta_1} = -1$; hence, $\dfrac{\partial z}{\partial \beta_1} = 1$

15.8 WORD LENGTH IN A/D CONVERTERS, MEMORY, ARITHMETIC UNIT AND D/A CONVERTERS

(*a*) Word length of A/D converter (open loop)

When a microcomputer is used to generate the control law, the control signal must be available in digital form. But most of the process variables are analog in nature and an analog to digital converter (ADC) is used to convert the analog signal to digital form for computer processing. The ADC introduces quantization noise.

The digital word length C + 1 (magnitude + sign) of the ADC is determined by the following two factors.

(*i*) Dynamic range of the incoming analog signal and

(*ii*) The quantization noise.

We consider the following example to illustrate the point.

Example 15.3: (*i*) The dynamic range is determined by the ratio between the maximum value e_{max} of the analog signal to its minimum value e_{min}. Assuming that e_{max} = 1 the quantization level, $q = e_{min} / e_{max} = 2^{-C}$ is the least significant bit solving for C we obtain

$$C = \log_2 \frac{e_{max}}{e_{min}} \tag{15.43}$$

(*ii*) The influence of the quantization noise of the ADC

Let C be the number of bits including sign bit for the ADC and C′ = C – 1 represent the magnitude part. So with truncation operation,

$$\sigma_{\varepsilon_t}^2 = \frac{q^2}{12} = \frac{4.2^{-2C}}{12} = \frac{2^{-2C}}{3}, \text{ where } q = 2^{-C'} = 2^{-(C-1)} = 2.2^{-C}$$

Assuming analog signal as a random Gaussian signal with a mean of 0.5 and maximum amplitude of unity (corresponding to 3σ). The variance of the analog signal is

$$\sigma_e^2 = \frac{1}{9}$$

Therefore, the signal to noise ratio will be

$$\frac{\sigma_e^2}{\sigma_{\varepsilon_t}^2} = \frac{1/9}{\dfrac{2^{-2C}}{3}} = \frac{1}{3} 2^{2C}$$

In decibel, $$F = 10 \log_{10} \frac{2^{2C}}{3} = 20 \text{ C} \log_{10} 2 - 10 \log_{10} 3$$

Hence, the number of bits, C of the ADC for a prescribed noise figure F can be solved as:

$$C = \frac{F + 10 \log_{10} 3}{20 \log_{10} 2} = \frac{F}{6} + 0.8 \tag{15.44}$$

ADC word length is, therefore, determined by the largest value of C from (15.43) and (15.44)

$$C \geq \max \left\{ \left[1 + \log_2 \frac{e_{max}}{e_{min}} \right], \left[\frac{F(dB)}{6} + 0.8 \right] \right\} \tag{15.45}$$

Example 15.4 Find the bit size of an A/D converter if the dynamic range of the analog input is such that the maximum saturation to minimum threshold is 250:1 Also the noise level should be less than 40 *d*B.

Here, $\frac{e_{max}}{e_{min}} = \frac{\text{Saturation value}}{\text{Threshold}}$ = 250 and F = 40 *db*

Therefore, C ≥ max [(1 + 8), (7.47)] ≥ 9 bits.

(*b*) The arithmetic logic unit

We know the signal to noise ratio is given by: $F = 10 \log_{10} \frac{\sigma_e^2}{\sigma_u^2}$

Let $\sigma_u^2 = K_m \sigma_e^2$, where K_m is found from residues of $D(z) D(z^{-1}) z^{-1}$ at poles of $D(z)$

Assuming the analog signal as before, we have, $F(db) = 10 \log_{10} \left[\frac{1/9}{\dfrac{2^{-2C} K_m}{3}} \right]$

$$= 10 \log \frac{2^{2C}}{3K_m} = 20\ C \log_{10} 2 - 10 \log_{10} 3 - 10 \log_{10} K_m \tag{15.46a}$$

Therefore, $C \geq \frac{F}{6} + 0.79 + 1.66 \log_{10} K_m$

Example 15.5 Find the required bit size for an arithmetic logic unit for implementing the controller D(z) given below so as to maintain a signal to noise ratio of 45 dB at the controller output.

$$D(z) = \frac{1}{1 - 0.95 z^{-1}}$$

Here, K_m is found as: $K_m = \frac{1}{1-\beta^2} = \frac{1}{1-(0.95)^2} = 10.2564,$

Since, F = 45 db the bit size of arithmetic logic unit is found to be:

$$C \geq 7.47 + 0.79 + 1.66 = 9.94 \geq 10.$$

(*c*) Memory word length

We will estimate the influence of the co-efficient word length on pole relocations. Let the maximum permitted relocation of the pole P in the s-plane be represented by ΔP. The corresponding relocation on z-plane is :

$$\Delta z = e^{-(P - \Delta P)T} - e^{-(P + \Delta P)T} = 2e^{-PT} \sinh (\Delta P)T$$

The least significant bit 2^{-C} should be equal to Δz, Hence

$$2^{-C} = 2e^{-PT} \sinh (\Delta PT) \approx 2e^{-PT} \Delta PT$$

$$= PT\ e^{-PT} \cdot \frac{2\Delta P}{P}$$

$$\Rightarrow \qquad -C = \log_2 PT\ e^{-PT} + \log_2 \frac{2\Delta P}{P}$$

$$\therefore \qquad C = -\log_2 PT\ e^{-PT} - \text{Log}_2 \frac{2\Delta P}{P} \tag{15.46b}$$

$$= -3.32 \log_{10} PT\ e^{-PT} - 3.32 \log_{10} \frac{2\Delta P}{P}$$

Example 15.6 : For a pole located at P = 2.0 rad/sec and sampling period of T = 0.005 second, find the memory word length to attain a maximal pole sensitivity of 2%.

Now, using relation (15.46b) we get

$$C \geq 6.65 + 4.64 > 11 \text{ bits}$$

Therefore, the bit size C + 1 needed to meet the specification, is given by : $C + 1 \geq 12$ bits.

(*d*) Closed loop considerations of the word length

Often the following rules of thumb are followed in selecting the bit sizes of various components

$(C + 1)_{A/D}$ is determined by the allowable limit cycle.

$$(C + 1)_{\text{Arithmatic}} = 4 + (C + 1)\ A/D \tag{15.47}$$

$$(C + 1)\ D/A = (C + 1)_{ADC} - 2 \tag{15.48}$$

15.9 QUANTIZATION GIVES RISE TO NON-LINEAR BEHAVIOR IN CONTROLLER

Another source of error is a slow varying quantization error which is nonlinear. A recursive filter implemented on a microcomputer with finite word length represents a nonlinear feedback system which, in turn, generates nonlinear effects like Dead band and limit cycle. We will demonstrate the dead band and limit cycle on decimal arithmetic. We will use an equivalent 2's complement for truncation *e.g.,* 8.6 truncated to decimal point becomes 8 and – 8.6 becomes – 9.

(*a*) Dead zone

Implementation of a digital controller in a finite bit computer may cause a steady state error in the control system. Consider a first order controller given by

$$\frac{m(z)}{e(z)} = \frac{1}{1+b_1 z^{-1}} \text{ so that } m(k) = -b_1 m(k-1) + e(k) \tag{15.49}$$

Let us examine the behavior of the digital controller implemented on a 3 bit computer with $b_1 = -0.6$. Assume that at time zero ($k = 0$) an impulse $e(k)$ of strength 4/8 is applied. The controller output for an infinite bit computer would decay to zero and force the process output to the desired value in the closed loop configuration, since

$$m(\infty) = \lim_{z \to 1}(1 - z^{-1})m(z) = \lim_{z \to 1} \frac{z-1}{z} \frac{z}{z+0.6} \frac{4}{8} = 0$$

However, with the round-off quantizer in the loop, the result is shown in Table 15.2

Table 15.2

k	$m(k-1)$	$0.6m(k-1)$	$Q_r^3\{0.6m(k-1)\}$	$m(k)$
0	0	0	0	4/8
1	4/8	2.4/8	2/8	2/8
2	2/8	1.2/8	1/8	1/8
3	1/8	0.6/8	1/8	1/8
4	1/8	0.6/8	1/8	1/8
5	1/8	0.6/8	1/8	1/8

It is found from the Table 15.2, that the signal $m(k)$ never reaches value of zero as in the ideal case and it is exhibiting a dead band characteristic., We can show, by proceeding, in the same way, that a truncation quantizer will produce a zero value for $m(3)$.

(*b*) Limit cycle due to overflow in the controller output

A limit cycle is a condition of sustained oscillation in a closed loop system caused by nonlinearities within the loop.

Consider the transfer function of a compensator as given below :

$$\frac{Y(z)}{M(z)} = \frac{M(z)}{U(z)} \cdot \frac{Y(z)}{M(z)} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{15.50}$$

such that

$$\frac{Y(z)}{M(z)} = a_0 + a_1 z^{-1} + a_2 z^{-2} \text{ and } \frac{M(z)}{U(z)} = \frac{1}{1 + b_1 z^{-1} + b_2 z^{-2}} \tag{15.51}$$

Further, consider the numerical values $b_1 = 1.5$ and $b_2 = 0.56$ yielding roots for the characteristic equation $1 + 1.5\,z^{-1} + 0.56^{-2} = 0$ located at $z_1 = -0.8$ and $z_2 = -0.7$.

We are interested to analyze the effects of finite bit size in the implementation of the controller in Equation (15.51). Equation (15.51) can be written in discrete time domain as :

$$m(k) = -b_1 m(k-1) - b_2 m(k-2) + u(k) = -1.5m(k-1) - 0.56m(k-2) + u(k) \quad (15.52)$$

Let us consider a 4 bit computer for ease of hand calculation and for illustrative purpose with sign magnitude number representation and let the input be an impulse applied at $k = 0$, with $u(0) = 4/8$. We compute the value of $m(k)$ using relation (15.52) with rounding operation and present the result in Table 15.3 for $k = 0$ to 8.

We observe that the entries in the Table 15.3 corresponding to $k = 5$ and $k = 7$, are identical, which means that $m(k)$ is exhibiting a limit cycle with amplitude of oscillation $-1 \le m(k) \le 1$.

If we repeat the computation with a different impulse input of $u(0) = 5/8$ and present the result as in Table 15.4, we can see from the table that $m(k)$ exhibits limit cycle oscillation with amplitude $-1 \le m(k) \le 0.75$ due to overflow at $k = 2$. The amplitude of oscillation has decreased even though the impulse of increased strength was the input.

Note the following for checking the validity of the entries in the column corresponding to $m(k)$:

We can write 12/8 = 7/8 + 5/8; In binary 7/8 = 0.111 and 5/8 = 0.101;

Therefore, 7/8 + 5/8 = 1.100 = – (0.100) = – (1×2^{-1}) = – 1/2 = – 4/8 = 12/8;

Hence, 12/8 – 3/8 = – 4/8 – 3/8 = – 7/8 (= 9/8)

Table 15.3 Response of Equation (15.52) with impulse strength u(0) = 4/8

k	$m(k-1)$	$Q_r^3\{-1.5m(k-1)\}$	$m(k-2)$	$Q_r^3\{-0.56m(k-2)\}$	$m(k)$
0	0	0	0	0	4/8
1	4/8	– 6/8	0	0	– 6/8
2	– 6/8	9/8	4/8	– 2/8	7/8
3	7/8	– 11/8	– 6/8	3/8	– 8/8
4	– 8/8	12/8	7/8	– 4/8	8/8
5	**8/8**	**– 12/8**	**– 8/8**	**4/8**	**– 8/8**
6	**– 8/8**	**12/8**	**8/8**	**– 4/8**	**8/8**
7	**8/8**	**– 12/8**	**– 8/8**	**4/8**	**– 8/8**
8	– 8/8	12/8	8/8	– 4/8	8/8

Table 15.4 Response of Equation (15.52) with impulse strength u(0) = 5/8

k	$m(k-1)$	$Q_r^3\{-1.5m(k-1)\}$	$m(k-2)$	$Q_r^3\{-0.56m(k-2)\}$	$m(k)$
0	0	0	0	0	5/8
1	5/8	– 8/8	0	0	– 8/8
2	– 8/8	12/8	5/8	– 3/8	9/8 → – 7/8
3	**– 7/8**	**11/8**	**– 8/8**	**4/8**	**15/8 → – 1/8**
4	**– 1/8**	**2/8**	**– 7/8**	**4/8**	**6/8**
5	**6/8**	**– 9/8**	**– 1/8**	**1/8**	**– 8/8**
6	**– 8/8**	**12/8**	**6/8**	**– 3/8**	**9/8 → – 7/8**
7	**– 7/8**	**11/8**	**– 8/8**	**4/8**	**15/8 → – 1/8**
8	– 1/8	2/8	– 7/8	4/8	6/8

15.10 AVOIDING THE OVERFLOW

15.10.1 Pole Zero Pairing

There are a number of ways to eliminate the limit cycle oscillation due to overflow in a digital controller. When writing the controller transfer function in the form of equation (15.12), (15.13) and (15.14) pairing of the poles and zeros according to the steps given below will help in avoiding overflow. Even though it is not optimal [69], it gives good results by minimizing the response of pole pairs that are near the unity circle.

Step 1. Plot poles and zeros graphically in the z-plane

Step 2. *Pair real poles with each other:* Find the real pole nearest the $z = +1$ point, pair it with the real pole furthest from the $z = +1$ point. Continue until all real poles have been paired.

Step 3. *Pair complex poles and zeros:* Find the pole nearest the unit circle, match it with the zero nearest its location. If the zero is real, match the other pole of the pole pair in the same manner with the nearest zero. Repeat step 3 until all the finite poles and zeros are matched.

Example 15.7 Let β and α represent the poles and zeros, respectively, of a digital controller as shown below.

$$z \pm j\alpha_1 = 1 + 1.2z^{-1} + 0.5z^{-2}, \qquad z \pm j\beta_1 = 1 + 0.8z^{-1} + 0.7225z^{-2},$$

$$z + \alpha_2 = 1 + 0.5z^{-1}, \qquad z \pm j\beta_2 = 1 + 1.0z^{-1} + 0.89z^{-2},$$

$$z \pm j\alpha_3 = 1 - 1.5z^{-1} + 0.7225z^{-2}, \qquad z + \beta_3 = 1 - 0.4z^{-1}$$

The poles and zeros are plotted in the z-plane as shown in Fig. 15.14. We employ the following pairing strategy :

1. real pole β_3 with real zero α_2
2. complex pole β_1 is matched with complex zero α_1
3. complex pole β_2 is matched with complex zero α_3

Therefore, $D_i(z)$, i = 1, 2, 3 in Equation (15.11) become

$$D_1(z) = \frac{z+\alpha_2}{z+\beta_3},\ D_2(z) = \frac{(z+j\alpha_1)(z-j\alpha_1)}{(z+j\beta_1)(z-j\beta_1)} \text{ and } D_3(z) = \frac{(z+j\alpha_3)(z-j\alpha_3)}{(z+j\beta_2)(z-j\beta_2)}$$

Hardware or software implementation should be done based on this pairing of the poles and zeros for avoiding overflows.

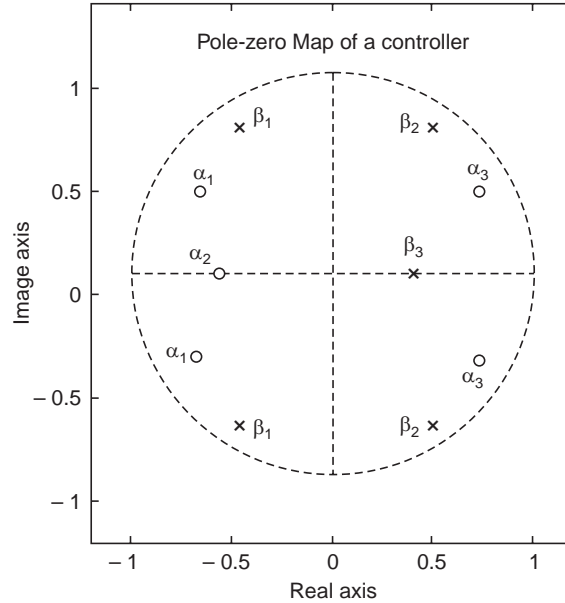



Fig. 15.14 Pole zero pairing to avoid controller overflow

15.10.2 Amplitude Scaling for Avoiding Overflow

Amplitude scaling may also be used to avoid the overflow at the controller output. If we consider a second order controller with transfer function of the form of relation (15.51) with complex conjugate poles, the response to a step input $u(k)$, in the worst case with zero damping ($b_1 = 0$) will make the magnitude of $m(k) = 2\,|\,u(k)\,|$, so that when $m(k)$ is scaled by dividing it with $2|\ u(k)\ |$, the normalized magnitude of $m(k)$ will never exceed the dynamic range of finite bit computer for any other non zero value of the damping ratio.

15.10.3 Design Guidelines

The following procedure may be used as guide lines for the implementation of cascade of second-order modules to achieve a 'good' sub-optimal digital controller.

1. Factorize the controller transfer function D(*z*) of (15.53) to express it in the form of Equation (15.54)

$$D(z) = \frac{\sum_{i=0}^{n} a_i\, z^{-1}}{1 + \sum_{i=1}^{n} b_1\, z^{-1}} \tag{15.53}$$

$$= \frac{\prod_{i=1}^{m} \alpha_i(z)}{\prod_{i=1}^{m} \beta_i(z)} \tag{15.54}$$

where $\alpha_i(z) = \alpha_{i0} + \alpha_{i1}z^{-1} + \alpha_{i2}z^{-2}$

$\beta_i(z) = 1 + \beta_{i1}z^{-1} + \beta_{i2}z^{-2}$

2. Quantize the coefficients for a desired word length (*e.g.,* 16 bits for the INTEL 8086)
3. Check that the controller in Equation (15.54) with quantized coefficients meets the system specifications.
4. Perform the pole zero pairing.
5. Select a structure for the controller modules (*i.e.,* Direct form 1 or direct form 2)
6. Introduce scaling to each module independently based on unit step method (provided if the forward path does not contain an integrator. However, any other methods may also be used.
7. Ordering of the modules in the chain may be simulated through MATLAB and Simulink or by some ordering algorithms [70]. This will complete the implementation.
8. Simulate the designed controller in the open loop configuration to ensure that the dynamic range of internal variables are in the mid range when subjected to a step input.
9. Simulate the implementation in the closed-loop case to ensure that system specifications are met.

The key to the foregoing procedure is that calculations are done on independent modules to determine scaling and ordering. This drastically reduces the computations needed to implement a filter.

MATLAB SCRIPTS

```
%Script_Example15.1
clear all; close all hidden;
num = [2 0.6 -0.2];               % Numerator polynomial in descending powers of z
den = [1 -0.65 0.1];              % Denominator polynomial in descending powers of z
z = roots(num)                    % zeros of the controller; for cascade realization
p = roots(den)                    % poles of the controller; for cascade realization
[R P K] = residue(num, den)       % Residues, poles and constant terms in partial
                                  % fraction expansion for parallel realization
```

```
%Script_Example15.4
clear all; close all hidden;
A=250;                                      % emax/emin
c1=1+log2(A);
F=40;                       %  Desired signal to noise ratio in dB
c2=(F+10*(log10(3)))/(20*log10(2)) ;  % Refer equation (15.44)
C3=max(c1,c2)                                % Refer equation (15.45)
C=round(C3)                        % Required minimum bit size for ADC
```

```
%Script_Example15.5
clear all; close all hidden;
F=45;              % Prescribed signal to noise ratio in dB
beta=0.95;         % pole location
beta1=power(beta,2);
K=1/(1-beta1);
C1= (F+10*(log10(3)+log10(K)))/(20*log10(2)); % See Equation 15.46a)
C = round(C1)      % Required minimum bit size for Arithmetic Logic Unit
```

```
%Script_Example15.6
clear all; close all hidden;
T= 0.005;     %Sampling time in Sec
P =2.0;       % Pole location
x1=P*T; x2=exp(-x1); x3=x1*x2; c1=-log2(x3)
y1=2*0.02;  % 2* delta p/P
c2=-log2(y1)
C3= c1+c2;         % Ref Equation 15.46b)
C=round(1+c1+c2)   % Required minimum bit size for memory
```

PROBLEMS

P15.1 The open loop plant transfer function of a system is given by

$$G_p(s) = \frac{10}{s(s+10)}$$

The desired closed loop system specifications are:

Damping ratio, $\delta = 0.5$

Settling time 1.2 sec

(*a*) Select a suitable a sampling period and find the equivalent z-plane roots along with the feedback coefficients in state feedback design.

P15.2 With reference to the block diagram of an antenna position control system shown in Fig. P15.2, the plant is an armature controlled dc motor with the following parameters:

Motor constant, KT = 1.000 Nm/A

Back emf, K_b = 1.2 V/rad/sec,

Armature inductance, L_a = 10 milli Henry

Armature resistance, R_a = 2.5 Ohm,

Moment of inertia of armature and that of reflected load, J = 0.0950 Kgm^2

Find the transfer function of the dc motor and design the analog compensator $G_c(s)$ to meet the closed loop system specifications :

Dominant pole damping ratio, $\delta = 0.7$

Band width, BW = 10Hz

Maximum pointing error in antenna position should be less than 0.25 deg with a disturbing force of F_d = 0.25 Nm.

Assuming a sampling frequency of f_s = 100 Hz, find the digitized version of the compensator with bilinear transformation and frequency prewarping and find the direct form controller coefficients represented on 8 bit microcomputer.

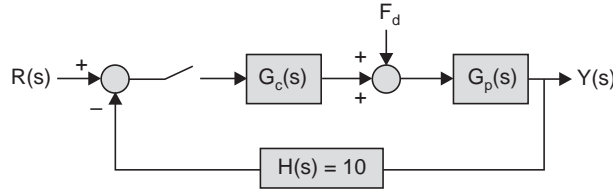



Fig. P15.2 System with disturbance

P15.3 Calculate multiplication noise generation of D(*z*) given below by using (*i*) parallel and (*ii*) direct form realization.

$$D(z) = \frac{0.1}{(1 - 0.9z^{-1})\,(1 - 0.95z^{-1})}$$

P15.4 Find the noise output for a compensator D(*z*) given by:

$$D(z) = \frac{1}{(1 - 0.5z^{-1})\,(1 - 0.9z^{-1})}$$

Find the block ordering that minimizes the output noise for cascade realization.

P15.5 Examine if limit cycle exits in a direct form realization for $D(z) = 1/(1 - 0.9z^{-1})^2$ when subjected to an impulse of 5/8.

P15.6 Find the variance of propagated noise at the output of the controller D(*z*) realized in cascade form as in (*a*) and realized in direct form as in (*b*). Compare the results found from (*a*) and (*b*).

(*a*) $$D(z) = \frac{(z + 1)^3}{(z - 0.9)\,(z - 0.85)\,(z - 0.8)\,(z - 0.75)}$$

(*b*) $$D(z) = \frac{z^3 + 3z^2 + 3z + 1}{z^4 - 3.3000z^3 + 4.0775z^2 - 2.2358z + 0.4590}$$

Mathematical Models of Some Representative Components and Systems

We shall derive the differential equation of some typical systems, which will constitute the components, or subsystems of a control system. From the linear differential equation, we can obtain either the Laplace transform of the equation and the transfer functions of the system or its state variable representation.

A.1 MERCURY THERMOMETER

Consider a mercury thermometer immersed in a flowing fluid for which the temperature θ_1 varies with time. We are interested to find out the response i.e., time variation of the thermometer reading θ_2. We make the following assumptions for the analysis.

1. The resistance to heat transfer offered by the glass and mercury is negligibly small compared to that of the film surrounding the bulbs.
2. All the thermal capacity is in the mercury whose temperature is uniform throughout its volume.
3. The glass wall containing the mercury does not expand or contract during the transient response.

It is assumed that before the thermometer was subjected to a temperature change, the thermometer was in a steady state.

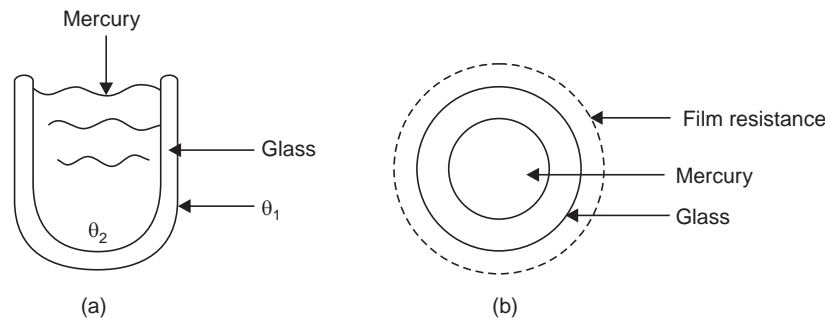



Fig. A.1. (*a*) Temperature gradients for a mercury thermometer (*b*) Cross-sectional view

Let A = Surface area of bulb for heat transfer (cm^2)

Cp = Heat capacity of mercury, Cal/gm°C

M = Mass of mercury in bulb, gm

t = Time, Sec

K = Film co-efficient of heat transfer, Cal/(sec) (cm^2)°C

By applying the unsteady-state energy balance :

input heat flow rate – output heat flow rate = rate of heat accumulation

we get $KA(\theta_1 - \theta_2) - 0 = M\,C_p(d\theta_2)/dt$ (A.1)

In Equation (A.1), θ_1, θ_2 are all functions of time, that is, $\theta_1 = \theta_1(t)$, and $\theta_2 = \theta_2(t)$. The explicit appearance of t will be omitted for clarity of expressions and is implied. It will be written explicitly as and when required. The Equation (A.1) states that the rate of flow of heat through film resistance surrounding the bulb causes the internal energy of the mercury to increase at the same rate.

The increase in internal energy is manifested by expansion of the mercury column and an increase in temperature. The co-efficient K will depend on the flow rate and properties of the surrounding fluid and dimension of the bulb but is assumed to be constant for a particular installation of thermometer. Equation (A.1) is a first order differential equation model of the *mercury-in-glass thermometer* and the transfer function can be calculated by taking its Laplace transform. We assume that the initial value of θ_2 is zero, so that after taking Laplace transform of Equation (A.1) the initial value term can be dropped. This means that θ_2 is now representing the incremental value of temperature in the bulb. Prior to the change in θ_2, the thermometer is at steady state and the derivative $d\theta_2/dt$ is zero. For steady state condition, Equation (A.1) becomes

$$KA\,(\theta_{1s} - \theta_{2s}) = 0 \qquad \text{for } t < 0, \tag{A.2}$$

The subscript s is used to denote the steady state temperatures, Equation (A.2) states that $\theta_{1s} = \theta_{2s}$, that is the thermometer reads the fluid temperature. Subtracting Equation (A.2) from Equation (A.1) we get

$$KA[(\theta_1 - \theta_{1s}) - (\theta_2 - \theta_{2s})] = MC_p\,\{d(\theta_2 - \theta_{2s})\}/dt = MC_p(d\theta_2/dt) \tag{A.3}$$

Let us use deviation variables of θ_1 and θ_2

$$\theta_1 - \theta_{1s} = u \qquad \text{and} \qquad \theta_2 - \theta_{2s} = y$$

So that Equation (A.3) becomes

$$KA(u - y) = MC_p dy/dt \tag{A.4}$$

or,

$$u - y = T dy/dt \tag{A.5}$$

where T = RC = thermal time constant, R = 1/KA = thermal resistance, C = MC_p = thermal capacity.

Taking the Laplace transform of Equation (A.5), we get

$$U(s) - Y(s) = TsY(s).$$

Therefore, the transfer function $G(s)$ between the deviation in input temperature u and the deviation in the output temperature $y = \theta_2 - \theta_{2s}$ of the thermometer is given by :

$$G(s) = \frac{Y(s)}{U(s)} = \frac{1}{1 + sT} \tag{A.6}$$

The parameter T has the dimension of time and is called the time constant of the thermometer. The time constant is a measure of the time taken by the system to adjust to a

change in input. The time constant of the thermometer is the product of the resistance R to heat flow through the film and the capacity C of heat storage in the mercury. In electrical circuits, the time constant is the product of electrical resistance and electrical capacitance and in a fluid-flow system the product of flow resistance and fluid-storage capacity.

In Equation (A.5) let $y(t) = x_1(t)$ be the only state in this first order system. With this choice, the Equation (A.5) can be recast as

$$u - x_1(t) = dx_1(t)/dt = \dot{x}_1(t) \quad \text{or,} \quad \dot{x}_1(t) = -x_1(t) + u(t) \tag{A.7a}$$

and
$$y(t) = x_1(t) \tag{A.7b}$$

which is the state variable representation of the first order system.

A.2 MIXING PROCESS

Consider a tank used to damp concentration fluctuations in the feed to a reactor (Fig. A.2). A stream of solution containing dissolved salt flows at a constant rate q into a tank of hold-up volume V. The concentration of the salt in the incoming stream u (mass of salt/volume), varies with time. We are interested to determine the transfer function relating the outlet concentration y to the inlet concentration u and to represent its dynamics in state variable form. With no chemical reaction and a constant level, the transient material-balance equation is

flow rate of salt in – flow rate of salt out = rate of accumulation of salt in the tank

Expressing this mass balance in terms of variables used in Fig. A.2 becomes:

$$q\text{U} - q\text{Y} = d(\text{VY})/dt) = \text{V}(d\text{Y}/dt) \tag{A.8}$$

where q is the constant flow rate of solution, m^3/sec, U is the concentration of salt, Kg/m^3, V = hold-up volume of Tank, Y is the outlet concentration of salt, Kg/m^3, t is time in sec.

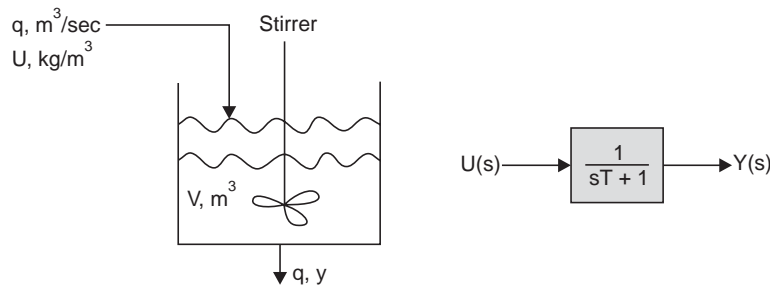



Fig. A.2 A blending tank

We shall again introduce deviation or incremental variables as we have done in the previous example. At steady state, Equation (A.8) may be written as

$$q\text{U}_s - q\text{Y}_s = 0 \tag{A.9}$$

Subtracting Equation (A.9) from Equation (A.8) we get

$$q(\text{U} - \text{U}_s) - q(\text{Y} - \text{Y}_s) = \text{V}\{d(\text{Y} - \text{Y}_s)/dt\} = \text{V}(d\text{Y}/dt)$$

or,
$$qu - qy = \text{V}(dy/dt) \tag{A.10}$$

where $u = \text{U} - \text{U}_s$, and $y = \text{Y} - \text{Y}_s$ are the deviation variables.

Taking the Laplace transform of the Equation (A.10) and remembering that the initial value of the deviation variable y is zero, we get

$$q\text{U}(s) - q\text{Y}(s) = \text{V}s\text{Y}(s)$$

or,
$$\frac{Y(s)}{U(s)} = \frac{q}{q + sV} = \frac{1}{1 + sT} \quad (A.11)$$

where the time constant T = (V/q), is the hold up time.

The assumptions underlying Equation (A.8) is that the concentration is uniform throughout the tank or that the input stream is instantaneously mixed with the liquid in the tank. Actually several seconds may be required for the complete mixing of the constituents in a tank and the output concentration is not the same as the average concentration of the tank when both are varying with time. For the present example, the hold up time is assumed to be at least two orders of magnitude greater than the mixing lag, which is neglected.

The state equation may be written from the Equation (A.10) by choosing $x_1(t) = y(t)$ as a state. So we have

$$\dot{x}_1(t) = -(1/T)\, x_1(t) + (1/T)\, u(t) \quad (A.12a) \qquad \text{and} \qquad y(t) = x_1(t) \quad (A.12b)$$

A.3 COUPLED LIQUID LEVEL SYSTEM

Let us consider the system in Fig. A.3 consisting of two tanks whose transfer functions cannot be obtained as the product of transfer functions of the individual systems. This is because of the fact that the tanks are not isolated, the presence of the second tank is affecting the dynamics of the first. This situation is described as the loading of the first unit by the second.

The variables in Fig. A.3 are defined as follows :

Q_s = steady state flow rate, m^3/sec
q = incremental input flow rate from steady state value in the first tank, m^3/sec
q_1 = incremental output flow rate from steady state value in the first tank, m^3/sec
q_2 = incremental output flow rate from steady state value in the second tank, m^3/sec
H_{s1} = steady state head in the first tank, m
h_1 = incremental change of head in the first tank, m
H_{s2} = steady state head in the second tank, m
h_2 = incremental change of head in the second tank, m.
C_1, C_2 = Capacitance of the first and second tank, respectively, m^2
R_1, R_2 = resistance to output flow rate in the first and second tank, respectively, $m/(m^3$/sec)

It is to be noted that the capacitance of a tank is defined as:

$$C = \frac{\text{change in liquid stored, } m^3}{\text{change in liquid head, } m}$$

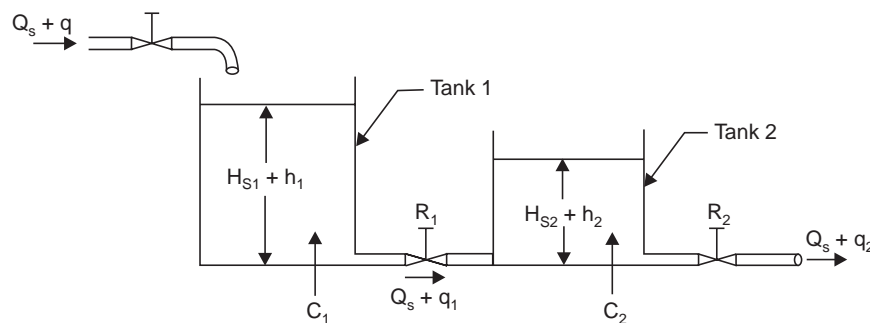



Fig. A.3 Two tanks coupled through a pipe

and the resistance to flow rate through a valve is defined as :

$$R = \frac{\text{differential of liquid heads across the valve, } m}{\text{incremental change of flow rate through the valve, m}^3\text{/sec}}$$

We assume that variation of flow rate q_1 and q_2 in the two tanks are very small compared to the steady flow rate Qs. Therefore, we can write the following relations using terminologies defined above (see Fig. A.3).

$$q_1 = \frac{h_1 - h_2}{R_1} \quad (A.13) \qquad q - q_1 = C_1 \frac{dh_1}{dt} \quad (A.14)$$

$$q_2 = \frac{h_2}{R_2} \quad (A.15) \qquad q_1 - q_2 = C_2 \frac{dh_2}{dt} \quad (A.16)$$

With $u = q$ as the input and q_2 as the output, the transfer function for the system in Fig. A.3 can be obtained from Equations (A.13) through (A.16) as

$$\frac{Q_2(s)}{U(s)} = \frac{1}{R_1C_1R_2C_2\, s^2 + (R_1C_1 + R_2C_2 + R_2C_1)s + 1} \quad (A.17)$$

One of the state space representations for the system may be obtained with the following choice of states, input and output respectively.

$$x_1(t) = h_2(t);\ x_2(t) = h_1(t);\ u(t) = q(t);\ y(t) = h_2(t); \quad (A.18)$$

Equations (A.16) can be recast as

$$C_2 \frac{dh_2}{dt} = \frac{h_1 - h_2}{R_1} - \frac{h_2}{R_2}; \qquad \text{or} \qquad \frac{dh_2}{dt} = -\left(\frac{1}{R_1C_2} + \frac{1}{R_2C_2}\right)h_2 + \frac{1}{R_1C_2}h_1$$

or

$$\dot{x}_1 = -\left(\frac{1}{R_1C_2} + \frac{1}{R_2C_2}\right)x_1 + \frac{1}{R_1C_2}x_2 \quad (A.19)$$

Similarly, recasting Equation (A.14) we get

$$C_1 \frac{dh_1}{dt} = q - \frac{h_1 - h_2}{R_1}; \quad \text{or} \quad \frac{dh_1}{dt} = \frac{1}{R_1C_1}h_2 - \frac{1}{R_1C_1}h_1 + \frac{1}{C_1}q \quad (A.20)$$

or

$$\dot{x}_2 = \frac{1}{R_1C_1}x_1 - \frac{1}{R_1C_1}x_2 + \frac{1}{C_1}u \quad (A.21)$$

So the state equation becomes

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix} -\left(\frac{1}{R_1C_2} + \frac{1}{R_2C_2}\right) & \frac{1}{R_1C_2} \\ \frac{1}{R_1C_1} & -\frac{1}{R_1C_1}\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{C_1}\end{bmatrix}u(t);\ y(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} \quad (A.22)$$

We will get a different state equation by making a choice :

$$x_1(t) = q_2(t),\ \dot{x}_1(t) = \dot{q}_2(t) = x_2(t);\ u(t) = q(t);\ y(t) = x_1(t)$$

With these choices and using Equations (A.13) through (A.16), we can write the state equation form as :

$$\begin{bmatrix}\dot{x}_1(t)\\ \dot{x}_2(t)\end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{1}{R_1C_1R_2C_2} & -\left(\frac{1}{R_2C_2} + \frac{1}{R_1C_1} + \frac{1}{R_1C_2}\right)\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} + \begin{bmatrix}0\\ \frac{1}{R_1R_2C_1C_2}\end{bmatrix}u(t);$$

$$y(t) = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}x_1(t)\\ x_2(t)\end{bmatrix} \quad (A.23)$$

So we see that even though there is a unique transfer function between q_2 and q, there may be a number of state variable models that describe the system dynamics.

A.4 ELECTRICAL CIRCUIT

Consider the simple RC circuit in Fig. A.4 and let us find out its mathematical model. The voltage source V(t) is connected in series with a resistance R and capacitance C. The voltage across the capacitor is the output of the circuit.

Let i be the current around the loop and applying the Kirchoff's Law around the loop we get

$$v(t) = R_i(t) + 1/C \int i(t)\, dt \qquad (A.24)$$

Recalling that the current is the rate of flow of charge (Coulombs per second), we may replace $i(t)$ by $dq(t)/dt$ in Equation (A.24) to obtain

$$v(t) = R(dq(t))/(dt) + \{q(t)\}/C \qquad (A.25)$$

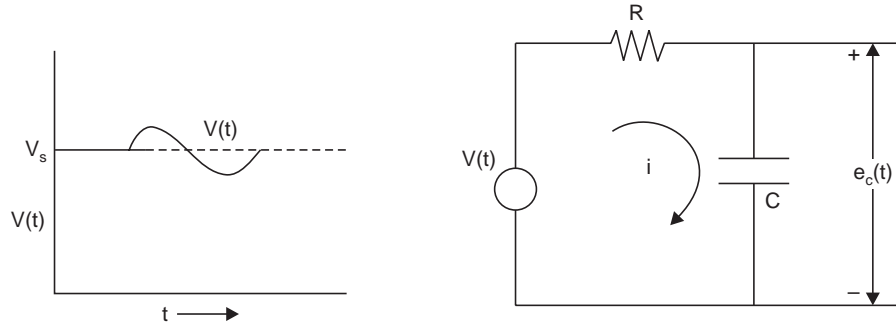



Fig. A.4 RC circuit

The voltage across the capacitance is given by

$$e_c = q(t)/C \qquad (A.26)$$

Combining Equations (A.25) and (A.26), we can write:

$$v(t) = R(dq)/dt + e_c \qquad (A.27)$$

Initially, when the circuit is at steady state dq/dt is zero and Equation (A.27) becomes

$$v_s = e_{cs} \qquad (A.28)$$

Subtracting Equation (A.28) from (A.27) and using deviation variables

$$V = v - v_s,\ Q = q - q_s,\ E_c = e_c - e_{cs} = Q/C$$

we can write the result as :

$$V = R(dq(t))/(dt) + Q(t)/C \qquad (A.29)$$

Taking the Laplace transform and rearranging we get

$$V(s) = RCs\, E_c(s) + E_c(s), \text{ where } E_c(s) = \{Q(s)\}/C$$

Therefore, the transfer function

$$G(s) = \frac{E_c(s)}{V(s)} = \frac{1}{sT+1}, \qquad \text{where } T = RC. \qquad (A.30)$$

Now with $Q(t) = x_1(t) = y(t)$, $V(t) = u(t)$, the state variable representation may be obtained from the Equation (A.29) as

$$\dot{x}_1(t) = -\,a\, x_1(t) + b\, u(t) \quad \text{(A.31}a\text{)}$$

$$y(t) = x_1(t) \quad \text{(A.31}b\text{)}$$

where $a = 1/\mathrm{T}$ and $b = 1/\mathrm{R}$

A.5 LINEARIZATION

All the examples considered thus far are linear, but most physical systems of practical importance are non-linear in nature. There is no problem as such in representing the dynamics of non-linear systems in state variable form but the transfer function model can be derived only for linear systems described by linear differential equations. The vast repository of analytical results available for linear systems using its transfer function, provides a motivation for obtaining an approximate linear representation of an actual nonlinear system by a technique known as Linearization. Such an example will now be considered.

A.6 LIQUID LEVEL SYSTEM

Consider the system shown in Fig. A.5(a), which consists of a tank of uniform cross-sectional area A to which is attached a flow resistance R such as valve or a pipe. The volumetric flow rate (volume/unit time) q_o through the resistance, is related to the head by the square-root relationship given by :

$$q_o(h) = \mathrm{C}\, h^{1/2}, \text{ where C is a constant} \quad \text{(A.32)}$$

Applying the material balance equation around the tank of uniform cross-sectional area, A and containing a liquid of constant density, we get

$$q(h) - q_o(t) = \mathrm{A}(dh)/dt \quad \text{(A.33)}$$

Combining Equations (A.32) and (A.33) we get

$$q(h) - ch^{1/2} = \mathrm{A}(dh)/dt \quad \text{(A.34)}$$

At this point we cannot proceed further and take Laplace Transform of the Equation (A.34). This is because of the fact that no simple transform exist for the non-linear term $h^{1/2}$. The difficulty can be overcome as follows :

By means of Taylor-series-expansion we can expand the function $q_0(h)$ around the steady value h_0 as:

$$q_o(h) = q_o(h_s) + q'_o(h_s)(h - h_s) + q''_o(h_s)\{(h - h_s)^2\}/2 + \,.\,.\,.\,.$$

where $q'_o(h_2)$ is the first derivative of q_o evaluated at h_s, $q_o''(h_s)$ is the second derivative of q_o etc.

Retaining only the first two terms,

$$q_o(h) = q_o(h_s) + q_o'(h_s)\,(h - h_s) \quad \text{(A.35)}$$

and substituting the value of q_0 from Equation (A.32) into Equation (A.35) yields

$$q_o(h) \cong q_o(h_s) + 1/2\ \mathrm{C}\ h^{-1/2}\,(h - h_{os})$$

or,
$$q_o(h) = q_{os} + \frac{h - h_s}{\mathrm{R}_1} \quad \text{(A.36)}$$

where $q_{os} = q_o(h_s)$ and $\dfrac{1}{\mathrm{R}_1} = \dfrac{d}{dh}\{q_o(h_s)\} = \dfrac{1}{2}\mathrm{C}\,h_s^{-1/2} = \dfrac{\mathrm{C}}{2\sqrt{h_s}}$

Substituting the value of $q_0(h)$ from Equation (A.36) in the Equation (A.34) we get

$$q(h) - q_{os} - (h - h_s)/\mathrm{R}_1 = \mathrm{A}(dh)/dt \quad \text{(A.37)}$$

At steady state dh/dt is zero and $h = h_s$ since the flow entering the tank equals the flow leaving the tank.

Hence, $$q(h_s) = q_s = q_{os} \quad \text{(A.38)}$$

Utilizing Equation (A.38) and introducing the deviation variables, $Q = q - q_s$, $H = h - h_s$, into Equation (A.37) gives

$$Q - \frac{H}{R_1} = A\frac{dh}{dt} \quad \text{(A.39)}$$

Taking the Laplace Transform we get,

$$Q(s) = \frac{H(s)}{R_1} + A\,s\,H(s)$$

or, $$\frac{H(s)}{Q(s)} = \frac{R_1}{1 + AR_1 s} = \frac{R_1}{1 + Ts} \quad \text{(A.40)}$$

where $T = AR_1$ and $R_1 = (2\sqrt{h_s})/C = (2\sqrt{h_s})/(q/\sqrt{h_s}) = 2h_s/q$

or $$q = 2h_s/R_1 = 2Ah_s/T$$

The resistance R_1 depends on the steady-state conditions around which the process operates. From Fig. A.5(*b*), we see that the resistance R_1 is the reciprocal of the slope of the curve passing through the operating point (q_{os}, h_s). It is also clear, from the graphical representation that, the Linearization of the nonlinear curve improves as the increment in h becomes smaller.

The validity of the linearized representation depends on the operation of the system. If the level of the liquid is maintained by a controller at or around the steady value h_s, then by the very nature of the control action the deviation in h is small from the steady height h_s and the linear approximation will be adequate. If, on the other hand, the level should change over a wide range, the linear approximation will give a poor representation of the system and no meaningful conclusion can be drawn from the analysis of such a linearized system. In such cases, more complicated methods, for the analysis of nonlinear systems should be resorted to.

Linearization techniques can also be applied to cases where the analytical expression of the nonlinearity is not available, but a graphical representation is known. The non-linear resistance is the reciprocal of slope of the tangent line passing through the operating point.

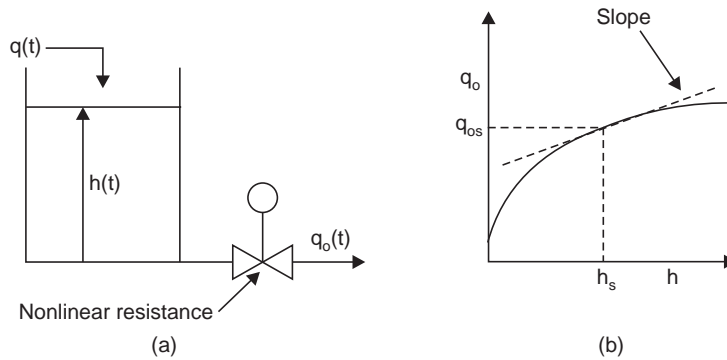



Fig A.5 Liquid-level system with non-linear resistance

In Equation (A.34), if we let $h(t) = y(t) = x_1(t)$ and $q(t) = u(t)$, then the state variable representation is obtained as :

$$\dot{x}_1(t) = -\,a_1(x_1(t)) + b\,u(t) \quad \text{(A.41}a\text{)}$$

$$y(t) = x_1(t) \qquad (A.41b)$$

where $a_1(x_1) = (C/A)\, x_1(t)^{-1/2}$ and $b_1 = 1/A$

For the linearized case $a_1(x_1) = (C/AR_1)$ where $\frac{1}{R_1} = \frac{d}{dh}\{q_o(h_s)\}$

A.7 TYPICAL MECHANICAL CONTROL SYSTEM DEVICES

The differential equation model of some typical mechanical control system devices will be derived in this section. The mathematical model includes the differential equation and the transfer function representation.

Mechanical devices can be classified as being either *transnational or rotational*. Newton's three laws of motion govern the action of both types of the systems. The only difference between the two is in the terminology we use for their description. Force, position, velocity and acceleration are the terms used in connection with transnational motion whereas, torque, angular position, angular velocity and angular acceleration are the corresponding terms for rotational motion. The laws of Newton states that the sum of the reactive forces or torque's, is zero for a body whose acceleration is zero. An alternative way of stating this is that the sum of the forces must be equal to zero for a body at rest or moving at a constant velocity. Some simple mechanical systems are considered for illustration.

A.8 MECHANICAL TRANSLATION SYSTEMS

The three basic quantities that characterize a mechanical transnational system are mass, damping and stiffness. Mass represents, a mechanical element having the property of inertia. Damping or viscous friction, represents a characteristic of an element that absorbs energy and stiffness represents the property of elasticity of an element like spring.

The symbols for the variables associated with transnational motion along with their units are shown in Table A.1. Before discussing the motion of mechanical systems involving more than one element, the force equation of the basic elements are considered first.

(*a*) Force applied to a mass

When a force, is applied to a fixed mass M, it produces an acceleration to the mass. A reaction force f_M is developed due to this acceleration in the direction opposite to the applied force. It is equal to the product of mass and acceleration (assuming constant mass)

Table A.1

Quantity	*Symbol*	*MKS units*
Distance	y	Meter, m
Velocity	$\dot{y} = (dy)/(dt)$	m/second.
Acceleration	$\ddot{y} = (d^2y)/(dt^2)$	m/second^2
Force	f	Newton, N
Mass	M	Kilo-Gram
Damping factor	B	N/m/second
Stiffness factor	K	N/m

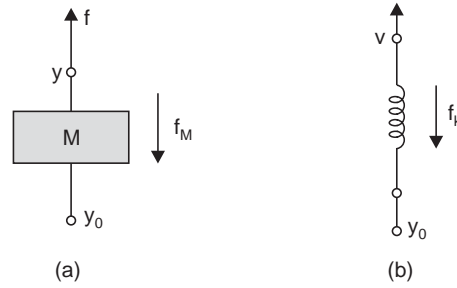



Fig. A.6 Force applied to (*a*) Mass M and (*b*) a Spring

In the Fig. A.6(*a*), y_0 is the displacement of the reference end and y is displacement of the mass. Therefore, the equation of the opposing force is given by

$$f_M = M \frac{d^2}{dt^2}(y - y_0) \tag{A.42}$$

If the reference is assumed to be stationary, f_M is given by

$$f_M = M \frac{d^2 y}{dt^2} \tag{A.43}$$

(*b*) Force applied to a spring

If a spring is stretched by an external force f, it tries to contract and if compressed, tries to expand to its normal length. In Fig. A.6(*b*) y and y_0 represent the position of the two ends of the spring measured from the equilibrium position and K is the stiffness factor. The restoring force f_K on the spring is given by Hook's Law as:

$$f_K = K(y - y_0) \tag{A.44a}$$

If the reference is selected such that $y_0 = 0$ Equation (A.45*a*) reduces to

$$f_K = Ky \tag{A.44b}$$

(*c*) Force applied to a damping device

A damping device has the characteristic that it absorbs energy. The schematic representation of damping action is a dashpot shown in Fig. A.7(*a*). An oil-filled dashpot, which is used in a system for adding damping is shown in Fig. A.7(*b*).

Under the influence of an external force f, as shown in Fig. A.7(*b*), the fluid flows around the piston from one side to the other. Sometimes a small hole is drilled in the piston to facilitate the flow of the fluid. The damping B depends on the dimension of the dashpot and the fluid used.

When a force f is applied to the dashpot, a reaction force, f_B is developed which equals the product of the damping factor B and the relative velocities of the two ends of the dashpot. The magnitude of the reactive force is given by

$$f_B = B\{(dy/dt) - (dy_0/dt)\} \tag{A.45a}$$

If the reference terminal is assumed to be stationary then $dy_0/dt = 0$, and Equation (A.45*a*) reduces to

$$f_B = B\frac{dy}{dt} = B\dot{y} \tag{A.45b}$$

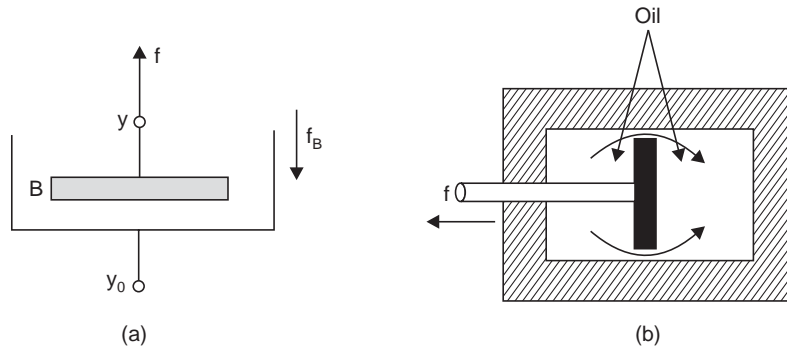



Fig. A.7 Damping device

A.8.1 Force Applied to a Mechanical System Containing Mass, Spring and Damping

In Fig. A.8(*a*), $y(t)$ represents the displacement of the mass measured from the reference, which is stationary due to an applied force $f(t)$. Since the spring is tied to the mass and the dashpot, the restoring force of the spring and the viscous friction can also be expressed in terms of y and its time-derivative, $\dot{y}$.

By applying Newton's Law, one gets $\qquad \mathrm{M}\,\ddot{y} + \mathrm{B}\,\dot{y} + \mathrm{K}\,y = f(t)$ (A.46)

Since, y represents incremental value, the Laplace Transform of this equation yields

$$\mathrm{M}s^2\,\mathrm{Y}(s) + \mathrm{B}s\mathrm{Y}(s) + \mathrm{KY}(s) = \mathrm{F}(s) \tag{A.47}$$

where Y(s) and F(s) are the Laplace Transform of $y(t)$ and $f(t)$ respectively.

The transfer function of this mechanical translational system, defined as the ratio of the Laplace Transform of the output Y(s) to that of the input, F(s), with zero initial conditions is given by

$$\frac{\mathrm{Y}(s)}{\mathrm{F}(s)} = \frac{\dfrac{1}{\mathrm{M}}}{s^2 + \dfrac{\mathrm{B}}{\mathrm{M}}s + \dfrac{\mathrm{K}}{\mathrm{M}}} \tag{A.48}$$

The block diagram of this system is shown in Fig. A.8(*b*).

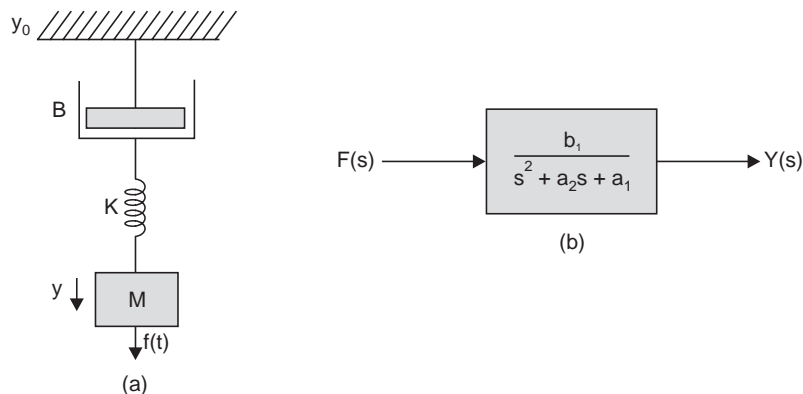



Fig. A8 (*a*) Mass-spring damping system (*b*) Transfer function

A state variable model of this system is obtained by setting $y(t) = x_1(t)$, $\dot{x}_1(t) = x_2(t)$, $f(t) = u(t)$ such that the Equation (A.48) can be written as:

$$\dot{x}_1(t) = x_2(t) \quad (A.49a)$$

$$\dot{x}_1(t) = -a_1 x_1(t) - a_2 x_2(t) \quad (A.49b)$$

where a_1 = K/M, a_2 = B/M, b_1 = 1/M. The state Equation (A.51) may be written in the matrix form

$$\dot{x}_1(t) = \text{A}\, x(t) + bu(t);\ y = cx(t) \quad (A.50)$$

where the system matrices A, b and c are given by, respectively

$$\text{A} = \begin{bmatrix} 0 & 1 \\ -a_1 & -a_2 \end{bmatrix},\ b = \begin{bmatrix} 0 \\ b_1 \end{bmatrix},\ c = [1 \quad 0]$$

A.8.2 Mechanical Lag-lead Network

A combination of springs and dampers as shown in Fig. A.9 may be used as a lag-lead compensator (vide Chapter 8) for mechanical positional control systems.

The equations of motion for the system in the figure are given by :

$$\text{B}_2(\dot{x}_1 - \dot{x}_2) + \text{K}_2(x_1 - x_2) = \text{B}_1(\dot{x}_2 - \dot{y})$$

$$\text{B}_1(\dot{x}_2 - \dot{y}) = \text{K}_1\, y \quad (A.51)$$

where K_i , B_i , i = 1, 2 are, respectively, the stiffness factor of the spring and damping factor of the dashpot and x_1, x_2, y are displacements indicated in the figure.

By taking the Laplace transforms of these two equations, with zero initial conditions, we obtain

$$\text{B}_2[s\text{X}_1(s) - s\text{X}_2(s)] + \text{K}_2[\text{X}_1(s) - \text{X}_2(s)] = \text{B}_1[s\text{X}_2(s) - s\text{Y}(s)]$$

$$\text{B}_1[s\text{X}_2(s) - s\text{Y}(s)] = \text{K}_1\,\text{Y}(s)$$

Eliminating Y(s) from the last two equations, the transfer function $\text{X}_2(s)/\text{X}_1(s)$ can be written as :

$$\frac{\text{X}_2(s)}{\text{X}_1(s)} = \frac{\text{K}_1\text{K}_2\left(\frac{\text{B}_1}{\text{K}_1}s+1\right)\left(\frac{\text{B}_2}{\text{K}_2}s+1\right)}{\text{B}_1\text{B}_2 s^2 + (\text{K}_1\text{B}_1 + \text{K}_1\text{B}_2 + \text{K}_2\text{B}_1)s + \text{K}_1\text{K}_2}$$

$$= \frac{(\text{T}_1 s+1)(\text{T}_2 s+1)}{\left(\frac{\text{T}_1}{\alpha}s+1\right)(\alpha\text{T}_2 s+1)} = \frac{\left(s+\frac{1}{\text{T}_1}\right)\left(s+\frac{1}{\text{T}_2}\right)}{\left(s+\frac{\alpha}{\text{T}_1}\right)\left(s+\frac{1}{\alpha\text{T}_2}\right)}$$

where $\text{T}_1 = \frac{\text{B}_1}{\text{K}_1},\ \text{T}_2 = \frac{\text{B}_2}{\text{K}_2},\ \frac{\text{B}_1}{\text{K}_1} + \frac{\text{B}_2}{\text{K}_2} + \frac{\text{B}_1}{\text{K}_2} = \frac{\text{T}_1}{\alpha} + \alpha\text{T}_2;\ (\alpha > 1).$

In order to use this mechanical network as a lag-lead compensator, it is desirable to add a suitable linkage with an adjustable gain K_c. With this variable gain the transfer function of the compensator becomes :

$$\text{G}_c(s) = \text{K}_c \frac{\left(s+\frac{1}{\text{T}_1}\right)\left(s+\frac{1}{\text{T}_2}\right)}{\left(s+\frac{\alpha}{\text{T}_1}\right)\left(s+\frac{1}{\alpha\text{T}_2}\right)} \quad \text{where } \alpha > 1. \quad (A.52)$$

We can obtain the state variable representation for the above system using Equation (A.51) or by following the procedure described in Section 3.5 of Chapter 3.

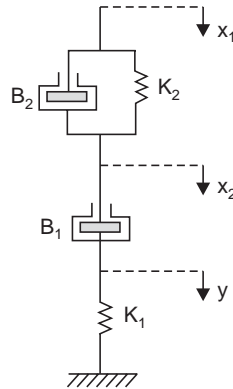



Fig. A.9 Mechanical lag-lead network

A.9 MECHANICAL ROTATIONAL SYSTEMS

The basic parameters that characterize a mechanical rotational system are moment of inertia, damping and stiffness or elastance. Before considering a mechanical rotational system containing more than one element, let us write first the torque equations of the basic rotational elements. Table A.2 shows the different symbols used in connection with rotational motion.

Table A.2

Quantities	*Symbol*	*MKS unit*
Angle	θ	Radians
Angular velocity	$\frac{d\theta}{dt} = \dot{\theta}$	Radians/second
Torque	T	Newton-Meter
Moment of inertia	J	Kg-meter^2
Damping factor	B	Newton-meter/(radian/second)
Stiffness factor	K	Newton-meter/radian

(*a*) Torque applied to a body possessing a moment of inertia

When an external torque is applied to a body having moment of inertia, J it produces angular acceleration and reactive torque is developed in the direction opposite to the applied torque. The reactive torque T_j is equal to the product of the moment of inertia J and the angular acceleration $\ddot{\theta}$. In Fig. A.10, θ represents the angular position of the rotating body measured from the equilibrium position taken as the zero angular position. The reactive torque is therefore given by :

$$T_j = J\ddot{\theta} \qquad \text{(A.53)}$$

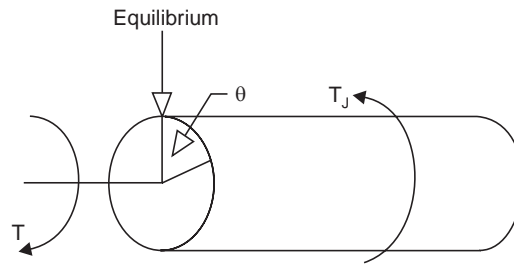



Fig. A.10 Torque applied to a body with moment of inertia J

(*b*) Torque applied to a damping device

An elementary damping device is shown in Fig. A.11 where a disc is rotated through some fluid, positions θ and θ_0 are the angular positions of the two ends of the damping device measured from the respective equilibrium positions. When a torque T is applied to the damping device, a reactive torque T_B is developed which is equal to the product of the damping factor and the relative velocities of the two ends of the damping device. It can be expressed as

$$T_B = B(\dot{\theta} - \dot{\theta}_0) \quad \text{(A.54)}$$

When the reference end is stationary $\dot{\theta}_0 = 0$ and Equation (A.54) reduces to

$$T_B = B\dot{\theta} \quad \text{(A.55)}$$

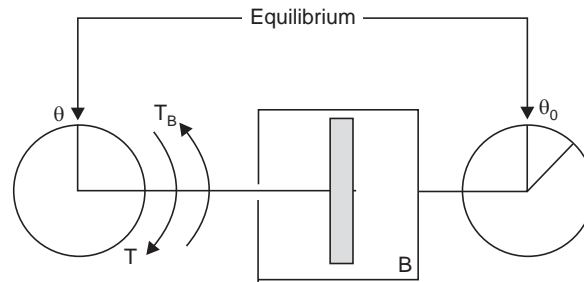



Fig. A.11 Torque applied to a damping device having damping factor B

(*c*) Torque applied to a twisting shaft

When an external torque T is applied to a shaft to produce a twist, the shaft tries to twist back with a reactive torque T_K due to its property of stiffness or elastance. In Fig. A.12,

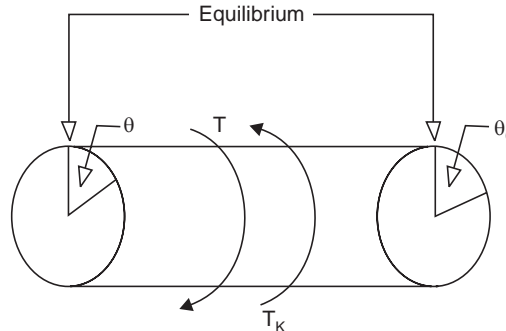



Fig. A.12 Torque applied to a twisting shaft with stiffness factor K

θ and θ_0 represent the angular positions of the two ends of the shaft measured from the equilibrium positions and K is defined as the stiffness factor. The restoring torque is given by

$$T_K = K(\theta - \theta_0) \tag{A.56}$$

When the reference end is stationary $\theta_0 = 0$ and Equation (A.56) simplifies to

$$T_K = K\theta \tag{A.57}$$

(*d*) Torque applied to a rotational system incorporating an element having moment of inertia, a twisting shaft and a damping device

The system in Fig. A.13 produces an angular displacement θ measured from an equilibrium position, which is assumed zero here, under the influence of an external torque T. Applying Newton's Law, this system yields

$$J\ddot{\theta}(t) + B\dot{\theta}(t) + K\theta(t) = T(t) \tag{A.58}$$

The Laplace Transform of the Equation (A.58) with zero initial conditions is found as :

$$Js^2\theta(s) + Bs\theta(s) + K\theta(s) = T(s)$$

Therefore, the transfer function of the system is given by :

$$\frac{\theta(s)}{T(s)} = G(s) = \frac{1}{Js^2 + Bs + K} = \frac{1/J}{s^2 + (B/J)s + K/J} \tag{A.59}$$

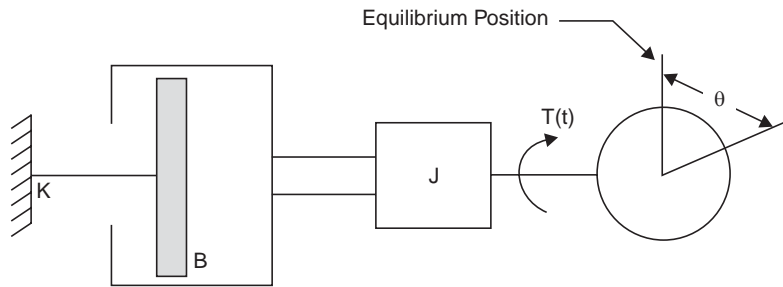



Fig. A.13 Mechanical system incorporating moment of Inertia, damping device and a twisting shaft

The block diagram of the transfer function in Equation (A.59) is shown in Fig. A.14.

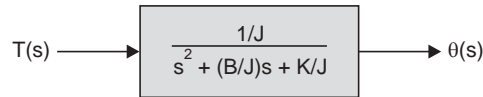



Fig. A.14 Transfer function of the system in figure A.13

The state variable representation of the form of Equation (A.50) is readily obtained with $x_1(t) = \theta(t)$ and $x_2(t) = \dot{\theta}(t)$ chosen as the two states for this second order system and torque $T(t) = u(t)$ as the input where $a_1 = K/J$, $a_2 = B/J$, and $b_1 = 1/J$.

Normally the mechanical load is connected to the motor shaft through gear trains (vide Fig. A.15). The moment of inertia, J, and viscous friction B in the above equations represent the effective values referred to the motor shaft and may be taken as:

$$J = J_m + n^2 J_L \tag{A.60}$$

$$B = B_m + n^2 B_L \tag{A.61}$$

where n (= $N_1/N_2 < 1$) is the gear ratio between the motor and load.

In relations (A.60) and (A.61) the moment of inertia and viscous friction coefficient of the gear train are assumed either negligible or included in J_L and B_L.

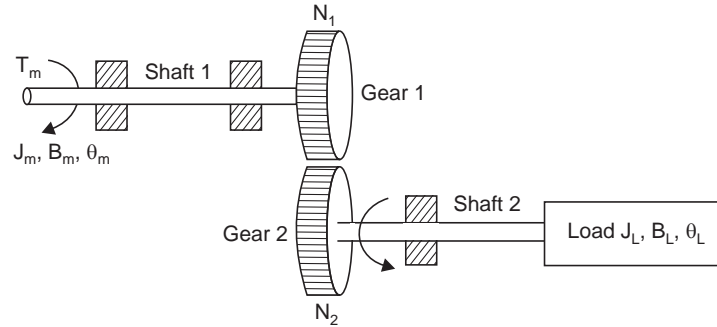



Fig. A.15 Load connected through gear train

A.9.1 DC Motors

Direct current machines are widely used in closed loop control systems, particularly for the control of speed and torque. They range in size from a few watts, driven by electronic amplifiers to several hundred kilowatts driven by Ward-Leonard generators or Amplidynes. DC motor used in control systems are generally separately excited.

Low power servo motors are normally used in instrumentation, particularly in aircraft control systems, where weight and space limitation require motors to deliver a maximum power output per unit volume. They are used for intermittent duty cycle or where large starting torques are required.

A current carrying conductor located in a magnetic field experience a force proportional to the magnitude of flux, the current, the length of the conductor, and the angle between the conductor and the direction of the flux. When the conductor is located at a fixed distance from an axis about which it can rotate, a torque is developed that is proportional to the product of the force and the radius. In a motor the resultant torque is the sum of the torques produced by the individual conductors. For a given rotor the only two adjustable quantities are the armature current and the flux. There are two modes of operation of a DC servomotor :

(*a*) Armature controlled mode and (*b*) Field controlled mode

In armature controlled mode, the field current is held constant and an adjustable control voltage is applied to the armature. In the field controlled mode, the armature current is held constant and an adjustable control voltage is applied to the field. By reversing the direction of control current, the direction of rotation of the motor is reversed. Two modes of operation are considered separately.

(*a*) Armature control

The field is separately excited by a constant current from a fixed DC source. A typical curve of the flux versus field current I_f is shown in Fig. A.16. As the field current increases to saturate the iron, the flux change is no longer proportional to the current change. In control systems, the operation is normally confined in the linear part of the characteristic, where the flux can be written as.

$$\phi = K_f I_f, \quad K_f = \text{constant} \tag{A.62}$$

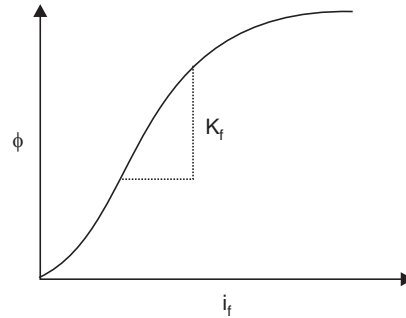



Fig. A.16 Flux density in the air gap with field current in a DC motor

The torque T developed by the motor is proportional to the product of ϕ and the armature current and the length of the conductors. Since the field is assumed to be constant, the developed torque for a given motor can be expressed as:

$$T = K_T \, i_a \tag{A.63}$$

where K_T = torque constant of the motor having units Nm/A.

The developed torque is used to drive the system having a total inertia J and to overcome the damping B and twisting the coupling shaft to the load. Neglecting the stiffness constant K, this can be expressed as :

$$T = J\ddot{\theta}_m + B\dot{\theta}_m \tag{A.64}$$

where θ_m is the angular position of the motor shaft.

As the armature rotates in a field, it develops an induced voltage e_b which is in a direction opposite to the armature supply. This voltage is called back *emf*. It is proportional to the speed of rotation $\dot{\theta}_m$ and the flux created by the field. Since the field is constant, the back *emf* can be expressed as

$$e_b = K_b \, \dot{\theta}_m \tag{A.65}$$

where K_b = voltage constant of the motor having units of volts/radian/second.

Control of the motor speed is obtained by adjusting the voltage applied to the armature. Its polarity determines the direction of rotation of the motor. A circuit diagram of the armature controlled DC motor is shown in Fig. A.17. The armature inductance, and resistance are designated by L_a and R_a respectively.

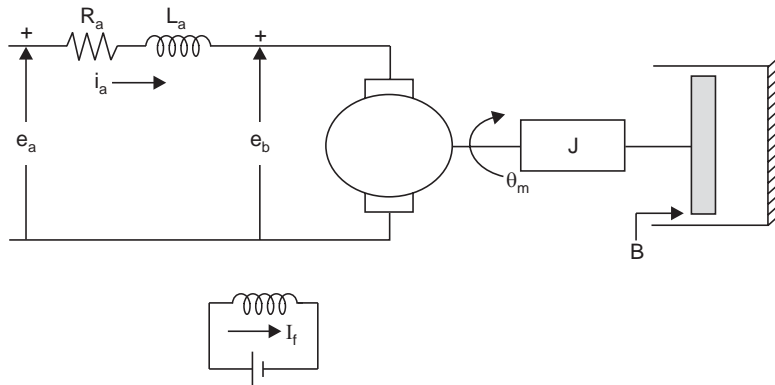



Fig. A.17 Armature controlled DC motor

The voltage equation of the armature circuit is

$$e_a - e_b = R_a i_a + L_a (di_a/dt) \tag{A.66}$$

Taking the Laplace Transform of the Equations (A.63), (A.64), (A.65) and (A.66) and setting the initial conditions to zero, the following relations can be obtained :

$$T(s) = K_T I_a(s) \tag{A.67}$$

$$\theta_m(s) = \{T(s)\}/\{s(sJ + B)\} \tag{A.68}$$

$$E_b(s) = K_b s\theta_m(s) \tag{A.69}$$

and

$$I_a(s) = \{E_a(s) - E_b(s)\}/(R_a + sL_a) = E(s)/(R_a + sL_a) \tag{A.70}$$

From these relations the block diagram of Fig. A.18 may be obtained.

The transfer function $\theta_m(s)/E_a(s)$ may now be obtained, either by block diagram reduction (see Section 2.7.1) from Fig. A.18 or by eliminating T(s), $E_b(s)$ and $I_a(s)$ from Equations (A.67) (A.68), (A.69) and (A.70), as

$$\frac{\theta_m(s)}{E_a(s)} = G(s) = \frac{K_T}{JL_a s^3 + (R_a J + L_a B)s^2 + (R_a B + K_b K_T)s} \tag{A.71}$$

The block diagram representation of Fig. A.18 clearly indicates the *presence of feedback* due to back e.m.f.

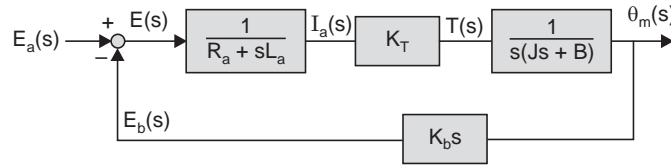



Fig. A.18 Block diagram for the system in figure A.17

Let $G_1(s)$ and H(s) represent the transfer function of the forward path and feed back path respectively.

Therefore, $H(s) = K_b s$

and

$$G_1(s) = \frac{K_T}{s(sL_a + R_a)(sJ + B)} = \frac{K}{s(s\tau_a + 1)(s\tau_m + 1)} \tag{A.72}$$

where $K = K_T/R_aB$, $\tau_a = L_a/R_a$ and $\tau_m = J/B$

The constants τ_a and τ_m have the dimension of time and are defined as the armature circuit time constant and motor time constant respectively. In DC motors the armature inductance can usually be neglected compared to armature resistance, which reduces the transfer function to :

$$G(s) = \frac{\theta_m(s)}{E_a(s)} = \frac{K}{s(s\tau_m + 1) + KK_b s} \tag{A.73}$$

The state variable representation of the armature controlled DC motor may be obtained from Equations (A.63), (A.64), (A.65) and (A.66) by selecting the states as $x_1(t) = \theta_m(t)$, $x_2(t) = \dot{\theta}_m(t)$, $x_3(t) = i_a(t)$ and the armature voltage $e_a(t) = u(t)$ as the input whereas the angular position as the output $y(t) = \theta_m(t) = x_1(t)$. The State equation matrices when expressed in the form of Equations (A.50) are given by :

Appendix A

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & \frac{-B}{J} & \frac{K_T}{J} \\ 0 & \frac{-K_b}{L_a} & \frac{-R_a}{L_a} \end{bmatrix}, \; b = \begin{bmatrix} 0 \\ 0 \\ \frac{1}{L_a} \end{bmatrix}, \; c = [1 \;\; 0 \;\; 0] \qquad \text{(A.74)}$$

(*b*) Field controlled DC motor

Fig. A.19 shows schematic diagram of the field controlled DC motor where the armature current is maintained constant and the field is supplied from the adjustable voltage e_f.

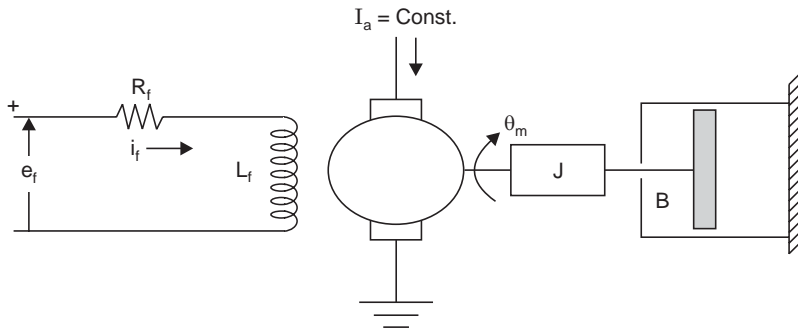



Fig. A.19 Field controlled DC motor

The torque, T developed by the motor is proportional to flux created by the armature current, the field current and the length of the conductors. For a given motor, and since the armature current is constant, the torque can be expressed as :

$$T = K_T \; i_f \qquad \text{(A.75)}$$

and its Laplace Transform becomes : $\quad T(s) = K_T \, I_f(s) \qquad$ (A.76)

where K_T is the torque constant (Kg-*m*/ampere).

This torque is utilized to drive a load of total inertia J and to overcome a viscous friction. This can be expressed, neglecting the stiffness constant K, as :

$$T = J(d^2\theta_m)/(dt^2) + B(d\theta_m/dt) \qquad \text{(A.77)}$$

Taking the Laplace Transform with zero initial conditions, it can be written as

$$T(s) = s(Js + B)\, \theta_m(s).$$

or

$$\theta_m(s) = \{T(s)\}/\{s(Js + B)\} \qquad \text{(A.78)}$$

Applying Kirchoff 's voltage law in the field circuit one gets

$$e_f(t) = R_f \, i_f + L_f(di_f)/dt \qquad \text{(A.79)}$$

and its Laplace transform yields : $E_f(s) = (R_f + sL_f)\, I_f(s)$.

or

$$I_f(s) = \{E_f(s)\}/(R_f + sL_f) \qquad \text{(A.80)}$$

Eliminating T(*s*), and $I_f(s)$ from Equations (A.76), (A.78) and (A.80) the overall transfer function $\theta_m(s)/E_f(s)$ can be obtained as

$$G(s) = \frac{\theta_m(s)}{E_f(s)} = \frac{K}{s(s\tau_f + 1)(s\tau_m + 1)} \qquad \text{(A.81)}$$

where $K = (K_T/R_f\, B)$, $\tau_f = L_f/R_f$ = field circuit time constant and $\tau_m = J/B$ is the motor time constant.

The block diagrams may be drawn utilizing the relations of Equation (A.76), (A.78) and (A.80) which will reveal that there is no inherent feed back present in field controlled DC motor.

The state variable representation is obtained by considering the output angular position and its derivative as the first two states, $x_1(t) = \theta_m(t)$, $x_2(t) = \dot{\theta}_m(t)$, the field current as the third state, $x_3(t) = i_f(t)$ and the field voltage as the input $u(t) = e_f(t)$ while the angular position is taken as the output $y(t) = \theta(t) = x_1(t)$. In this case the corresponding system matrices in the form of Equation (A.50) are given by:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -B/J & K_T/J \\ 0 & 0 & -R_f/L_f \end{bmatrix}, \; b = \begin{bmatrix} 0 \\ 0 \\ 1/L_f \end{bmatrix}, \; c = [1 \;\; 0 \;\; 0] \qquad (A.82)$$

A.9.2 AC Servomotor

The two phase AC servomotor is probably the most widely used type of servomotor in the power range of half-watt to 100 Watts. It is not too frequently used where larger power is involved because of its inherently poor efficiency. It fits very well as an actuator in a carrier frequency control systems. AC amplification can be used throughout the electrical portion of the system and since the AC servomotor can directly handle the amplitude modulated signal, demodulators can be dispensed with when the AC motor is used as the actuator.

The AC servomotor is basically two phase induction motor with a stator and a rotor. The stator consists of sheet metal laminations with slots to accommodate the phase distributed winding. The winding is distributed in such a way as to make the orientation of the axis of the two coils mutually at ninety degrees in space so that maximum useful torque is developed. The rotor construction may be a squirrel cage, solid or a drag-cup type. The most commonly used rotor is the squirrel cage, however, for fast responding instrument type servo-systems, drag cup rotors are used. The control signal is applied to one phase, referred to as *control phase*, while the other phase, referred to as *reference phase* is supplied from a fixed AC source, in order to produce torque efficiently, the phase voltages must be in *time-quadrature* along with the condition that the axis of the windings are in space quadrature. The 90 degree phase shift can be introduced either by connecting one capacitor in series with the reference-winding (Fig. A.20) or by suitable phase shifting network in the servo-amplifier.

The two windings are usually identical and equally rated. However, when the control power is at a premium and load power low, the reference winding may be rated at two to three times the control winding. It is essentially a constant speed motor and is operated either from 50 H*z* or 400H*z* source.

A schematic diagram of an AC servomotor driving a load of inertia J and damping B is shown in Fig. A.20, where e_c and e_r represent respectively the control phase and reference phase voltages.

The speed-torque characteristics of conventional two phase induction motor is shown as curve (*a*) in Fig A.21. Such a characteristic is entirely unsuitable for servo application owing to the fact that the slope of the curve is not negative over the entire speed range. Positive slope of the curve is an indication of negative damping which may easily lead to an undesirable condition known as instability.

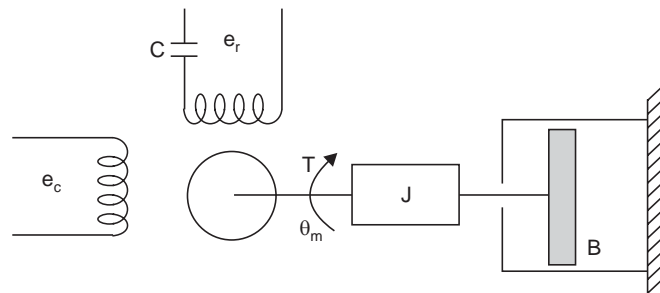



Fig. A.20 Control of two phase AC servomotor

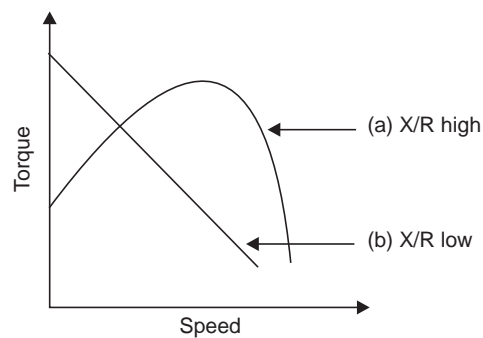



Fig. A.21 Characteristics of two phase AC servomotor

The ratio of the rotor reactance X to rotor resistance R is normally high for conventional motor and maximum torque is obtained at rated speed. In servomotor the ratio of X to R is decreased by increasing R and the speed-torque curve looks like curve (*b*) in Fig. A.21. The linearised torque-speed characteristic of a typical servomotor is shown in Fig. A.22 for various fixed values of control voltage.

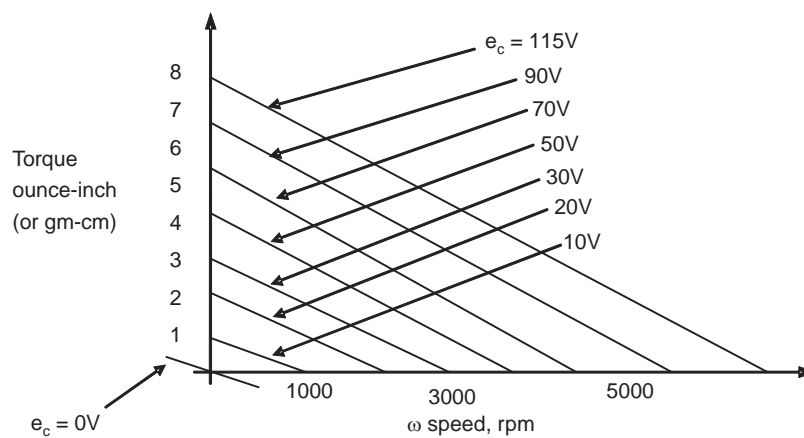



Fig. A.22 Linearised characteristic for 2-phase servomotor

It is to be noted that when the control voltage is zero, both the torque and speed are zero and since the slope is negative the motor will develop decelerating torque until it stops.

The curves also indicate that the stall torque is very high, which is desirable for producing rapid acceleration. Actual speed-torque curves are highly non-linear and linear differential equations are not adequate for their description. However, these curves may be approximated by straight lines and reasonable accuracy may be achieved (Fig. A.22).

The developed torque is a function of control voltage and speed, so the torque equation can be written in terms of partial derivatives as

$$T_q = f(\omega, e_c) = T_s(\omega_s, e_{cs}) + \frac{\delta T_q}{\delta\omega}(\omega - \omega_s) + \frac{\delta T_q}{\delta e_c}(e_c - e_{cs})$$

Therefore, $$T_q - T_s(\omega_s, e_{cs}) = -K(\omega - \omega_s) + K_1(e_c - e_{cs}) \qquad (A.83)$$

$$\text{where } \frac{\delta T_q}{\delta\omega} = -K, K > 0 \text{ and } \frac{\delta T_q}{\delta e_c} = K_1$$

Using incremental variables, $T = T_q - T_s(\omega_s, e_{cs})$, $\dot{\theta}_m = \omega - \omega_c$; $E_c = e_c - e_{cs}$

we can write $$T = -K\dot{\theta}_m + K_1E_c \qquad (A.84)$$

The Laplace Transform of the above equation is given by

$$T(s) = -Ks\theta_m(s) + K_1E_c(s) \qquad (A.85)$$

where $T(s)$, $\theta_m(s)$ and $E_c(s)$ are the Laplace Transforms of T, θ_m and E_c respectively.

The torque developed by the motor is utilized to drive a load of total inertia J and to overcome a viscous friction B. This can be expressed as :

$$T = J\ddot{\theta}_m + B\dot{\theta}_m \qquad (A.86)$$

The Laplace Transform of this equation, with zero initial conditions, is given by

$$T(s) = s(Js + B)\,\theta_m(s) \qquad (A.87)$$

The transfer function $\theta_m(s)/E_c(s)$ is obtained by eliminating $T(s)$ from the Equations (A.87) and (A.85)

$$\frac{\theta_m(s)}{E_c(s)} = G(s) = \frac{K_1}{s(sJ + B + K)} \qquad (A.88)$$

$$= \frac{K_m}{s(s\tau_m + 1)} \qquad (A.89)$$

where, $K_m = K_1/(B + K)$ = motor constant and $\tau_m = J/(B + K)$ = motor time constant.

A more exact analysis of the motor transfer function should include the effect of the lag in the control winding. The parameter for the speed-torque curve should be a function of control field current rather than control voltage. When analysed on this basis the transfer function of the motor angular displacement per unit control winding voltage can be shown to be of the form:

$$\theta_m(s)/E_c(s) = K_m/\{s(1 + s\tau_m)(1 + s\tau_f)\} \qquad (A.90)$$

where τ_f is the time constant of the control winding.

The value of τ_f is of the order of 0.001 to 0.005 sec and its contribution is practically negligible in comparison to other time constants and the transfer function of Equation (A.89) adequately describes the motor dynamics.

In the actual design problem, the numerical values of the constant in the expression of the transfer function are to be computed with the help of the manufacturer's data sheet. The detailed speed-torque characteristic is not supplied by the manufacturer. But the stall torque and the no load speed of the motor at rated voltages are only furnished. If the speed-torque

curves are assumed to be linear, the constants can be computed by remembering the following facts. In a closed loop control system, the motor normally operates around zero control for $e_c = 0$, where the slope is not the same as that of the curve for rated control voltage. The former is normally half the value of the latter.

Hence as a first approximation

$$K = -\frac{\delta T}{\delta \dot{\theta}_m}\Big|_{e_c = \text{constant}} = -\frac{1}{2}\frac{\text{stall torque at rated voltage}}{\text{no load speed at rated voltage}}$$

where the factor 1/2 accounts for the difference in slopes between the zero voltage and rated-voltage speed-torque curve. The other constant is

$$K_1 = \frac{\delta T}{\delta e_c}\Big|_{\dot{\theta}_m = \text{constant}} = \frac{\text{stall torque}}{\text{rated control voltage}}$$

The state variable representation for the linearised system can be readily written by using the Equations (A.84) and (A.86) with $\theta_m(t) = x_1(t)$, $\dot{\theta}_m(t) = x_2(t)$ and $E_c(t) = u(t)$.

Example A.1. Find out the transfer function of a two-phase servomotor driving a load of total inertia of 10 gm-cm^2 with negligible viscous friction, the stall torque of the motor is 1.8 ounce inch and the no load speed is 7500 rpm and the rated control voltage is 110 volts.

The constants are :

$K_1 = 1.8/110 = 1.6363 \times 10^{-2}$ ounce-inch/volt.

$K = -1/2\{1.8/(7500 \times 2\pi/60\} = -1.1459 \times 10^{-3}$ ounce-inch-sec

and $J = 10$ gm-cm^2 = (10/ 980) gm-cm-sec^2 = (10/980) × 0.0137 ounce-inch-sec^2

$= 1.39796 \times (10)^{-4}$ ounce-inch-sec^2

$\tau_m = J/K = 1.219966 \times 10^{-1} = 0.122$ sec,

motor constant $Km = K_1/K = 14.280$ (radian/sec)/volt.

The approximate transfer function : $G(s) = \dfrac{14.28}{s(0.122s + 1)}$

A.10 ELECTRIC SERVOMOTORS

Standard electric motors are not always suitable as *servomotors*. A servomotor is normally expected to produce rapid accelerations from near standstill or standstill conditions and this requires a motor with high starting torque and low inertia and the standard motor normally does not possess both of these features. A servomotor, therefore, *differs from standard motors principally* in that it has, in general, *high starting torque* and considerably *lower inertia*. Low inertia may be obtained in a number of ways, but the problem of maintaining high starting torque must be met simultaneously, for it is actually the ratio of torque to inertia that is important for servomotors. Servomotors are designed by using the following methods.

(*a*) The inertia of motor is decreased by reducing the diameter of armature. This also reduces the torque of the motor since the lever arm of the conductor is decreased. Increasing the length of the conductor of the armature compensates for the reduced lever arm and provides the required torque. If the diameter of the armature is reduced by a factor of 2 and the armature length is doubled, the torque to inertia ratio should increase, theoretically by a factor of 8.

(*b*) For small instrument DC motors (upto 30 watts), the moving part may be reduced to armature conductors alone, tremendously reducing the inertia. The field produced by a permanent magnet, the armature conductor rotate in the angular ring provided by a fixed central core and the field magnet.

(*c*) The torque of DC motors and AC commutator motors can be increased by using compensating winding so that greater peak control currents may be used.

(*d*) Very low inertia can be realized in AC induction motors by shaping the motor in the form of a cup and mounted in a frame consisting of fixed field and a fixed central core. This is Known as *drag-cup* motor.

Direct current motors are lighter than AC servomotor for the same power output, and have higher starting and reversing torques than AC motors. On the other hand, AC motors are extensively used in servo systems because of their economy, simplicity, reliability, fast response and they present no commutator problems. Of course, commutator problems may be overcome by using currently available brushless DC motors.

While selecting servomotor for a particular application the following points should be borne in mind.

For Selection of Servomotor:

1. The motor should have the required capacity to drive the load and overcome the losses associated with the gear trains.
2. The motor must produce the required amount of the acceleration as is demanded by the application.
3. The motor must be able to meet the maximum power requirement during transient conditions.
4. It must operate at a fixed velocity or a range of velocities.
5. The motor should be suitable for a specified duty cycle (continuous or intermittent).

A.11 TRANSDUCERS AND ERROR-SENSING DEVICES FOR MECHANICAL LOADS

In a feedback control system we need sensors and transducers to get information about the controlled variable and to set the reference input. However, the physical form of controlled variables may be diverse in nature and character, it can be a mechanical, electrical, hydraulic or thermal in nature. A glimpse of the areas where feedback control is used and the nature of variables controlled can be found in Section 1.5 of Chapter 1. A transducer is any device which converts energy from one form to another. A potentiometer is a transducer, since it converts mechanical shaft positions into proportional electrical signals (it is widely used as manual volume control in TV and radio). A tachometer generator is also a transducer, since it generates a voltage at its terminals proportional to the angular velocity (rpm) of the shaft to which its rotor is attached. Galvanometer, Linear Voltage Differential Transformer (LVDT), Synchro transmitter and Synchro Control Transformer are examples of electro mechanical transducers. A thermocouple is a familiar transducer that converts temperature to proportional voltage, a strain gauge is a transducer for converting pressure to change in current.

Transducers can be properly arranged to act as error-sensing devices for comparing two signals simultaneously. In feedback control systems, an error-sensing device is used to produce a signal which is proportional to the difference between the reference input and the controlled output variable. We shall now discuss, the use of some popular transducers as error sensing elements in feedback control systems.

A.11.1 Potentiometers as Error-sensing Elements

On a rectangular piece of electrically insulted flexible support (called mandrel), a length of oxide coated thin copper wire (copper oxide acting as insulator) is wound in the form of a coil (see Fig. A-23(*a*)). The mandrel, usually made from phenolic plastic, is then given a circular shape and is put into a case or housing made of insulating material like plastic (see Fig. A-23(*b*)). A wiper (or brush) is fitted to a rotating shaft to maintain electrical contact with the individual turns of the coil of copper wire acting as the resistance element. The wiper is provided with some spring action so that, when rotated, it remain pressed against a conducting concentric metal strip at one end and the resistance coil on the mandrel at the other end. The oxide coating of copper wire at the contact point on the coil where the brush is pressed against, is removed by sand paper or otherwise to facilitate electrical contact with the wiper. Electrical connection is then taken out from the conducting metal strip acting as the slip ring and the two free terminals of the resistance coil and attached to a binding post mechanically fixed to the housing. The schematic diagram of the potentiometer is shown in Fig. A.23(*c*). Potentiometers are also manufactured, where the wiper is made to move on a continuous concentric strip of carbon acting as a resistive element which, are found in manual-volume control in TV and Radio but are not preferred in servo applications. Potentiometers are manufactured in single turn, multiturn, or translatory to meet various requirements.

If the potentiometer characteristic is assumed to be linear as shown by the dotted line in Fig. A.23(*d*), the transfer function G(*s*) relating the output voltage V and the angular shaft rotation θ is given by

$$G(s) = \frac{V_0(s)}{\theta(s)} = K_s \quad (A.91)$$

The constant K_s is found from Fig. A.23(*d*) as

$$K_s = \frac{\text{Applied voltage across fixed terminals}}{\text{Maximum possible shaft rotation}} = \frac{V_R(s)}{\theta_{max}(s)}, \text{ volts/ radian} \quad (A.92)$$

The above relation implies that a single potentiometer can be used as an error sensing element for a mechanical load by connecting the reference input shaft to the potentiometer housing and the mechanical load attached to the shaft of the potentiometer. However, this scheme is feasible only when the mechanical load and the reference input shaft are physically located near to each other. However, the load and reference shaft are almost always remotely located. So, a pair of potentiometers, connected as shown in Fig. A-24 can serve as error sensor for such a situation.

The two potentiometers convert the input and output shaft positions into proportional electric voltages. The two voltages are, in turn, compared to generate the error voltage V_e between terminals *a* and *b*. The electrical circuit in Fig. A-24 represents a simple Wheatstone bridge, such that when the two arms are in the same relative position, the error voltage V_e is zero. If the position of the output shaft is above the input shaft as in Fig. A-24, the polarity of the error voltage is positive as the point ***b*** is at a higher potential than that of ***a***. The polarity of the error voltage will be opposite if the position of the load shaft is at a lower position with respect to the input shaft. The voltage source V of the bridge circuit in Fig. A.24 can also be replaced by an ac source. When the source voltage is dc, the polarity marks of V_e indicate the sign of the error voltage and if it is an ac voltage, the polarity marks refer to the phase of $V_e(t)$ with respect to that of the source voltage V. In either case, the transfer function of the error sensing device may be written as:

$$V_e = K_s(\theta_r - \theta_c) \quad (A.93)$$

where V_e = error voltage in volts

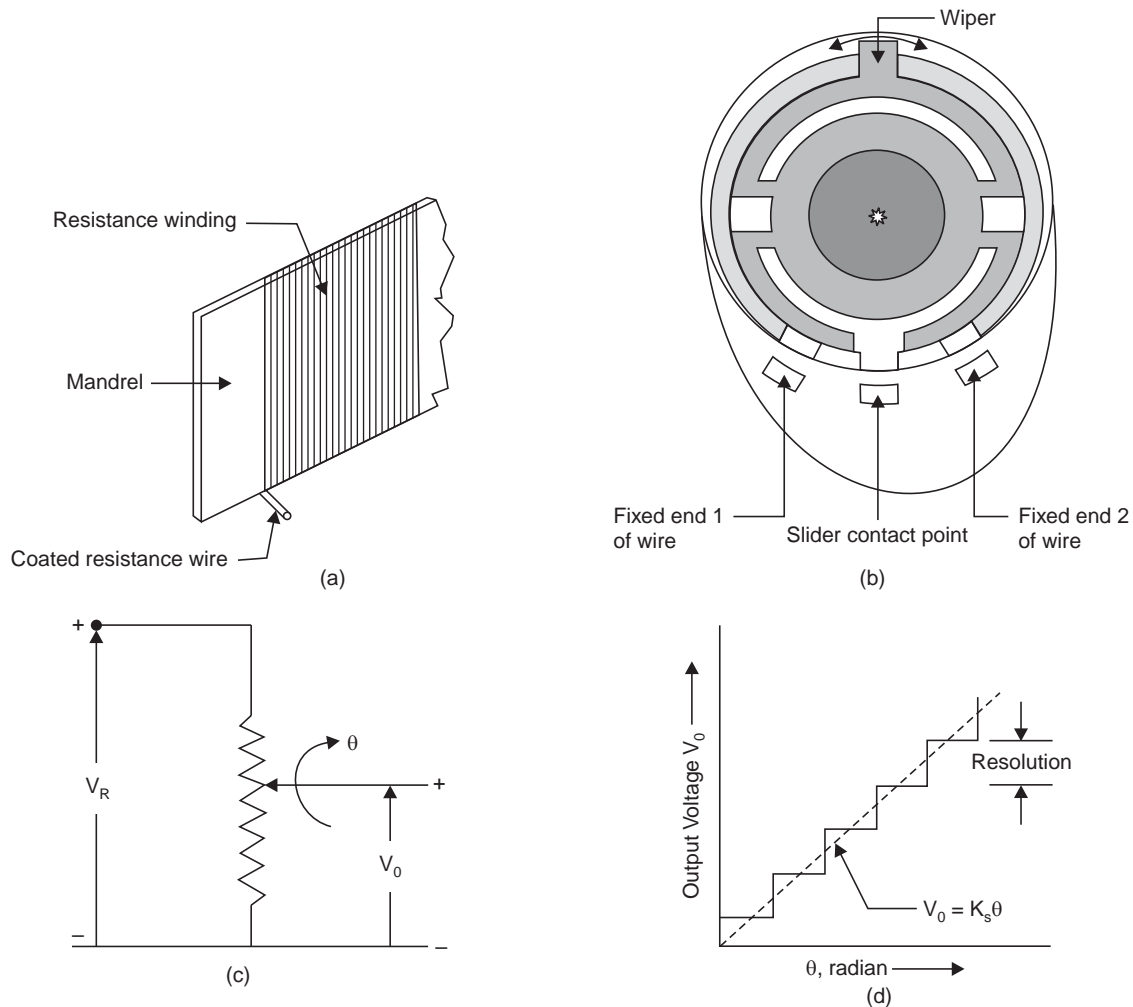



Fig. A-23 (*a*) Insulted copper wire wound on mandrel (*b*) a view of potentiometer with cover removed (*c*) Schematic diagram of potentiometer (*d*) Characteristic of wire-wound potentiometer

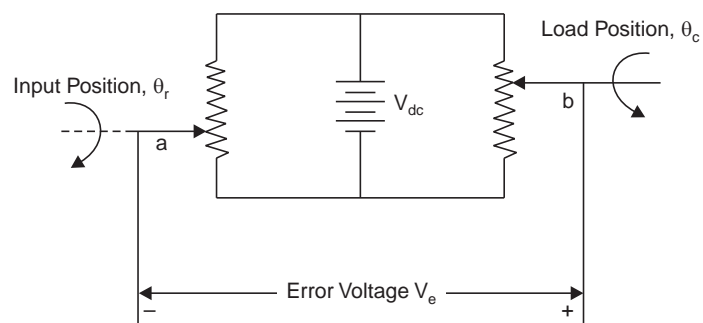



Fig. A-24: A pair of potentiometers used as error sensing elements

and K_s = sensitivity of the error detector in volts/rad

θ_r = reference input shaft position in radians

θ_c = controlled output shaft position.

A typical application of the pair of potentiometers as an error detector in a servomechanism has been shown in Fig. 4.4 in Chapter 4.

A.11.2 Resolution of Potentiometers

Most of the potentiometers used in servo applications are of the wire-wound type. As the potentiometer shaft is rotated, the sliding wiper contact moves from one turn of the coil to a neighbouring turn of the coil so the potential change only in discrete steps. Therefore, the change in output voltage of the potentiometer is not a continuous function of the rotation θ of the shaft, rather a staircase like variation is observed [vide Fig. A-23(*d*)].

The resolution of a potentiometer is defined as the minimum change in output voltage V obtained by rotating the shaft, expressed as a percentage of the total applied voltage V. If there are n number of turns of copper wire, the resolution can be expressed as :

$$\text{Resolution} = \frac{\Delta V}{V} = \frac{V/n}{V} = \frac{1}{n} \tag{A.94}$$

The resolution of a potentiometer places an upper limit on the accuracy of a servo system. The resolution of precision wire-wound potentiometers ranges between 0.001-0.5 per cent. For a carbon potentiometer, the resolution is zero, since the wiper moves along a continuous strip of carbon acting as the resistive element.

Although the principle of operation and construction of a potentiometer is quite simple, its application in feedback control system suffers from the following major shortcomings:

(1) Precision wire-wound potentiometers are very delicate devices. The sliding contact may be damaged if it is not handled carefully.

(2) The output voltage is discontinuous; causing inaccuracy in servo applications.

(3) The useable angle of rotation is less than 360 deg, for a single-turn potentiometer,

(4) The voltage applied to the potentiometer cannot be too large, because of the limit on heat dissipation of the potentiometer. The sensitivity, therefore, seldom exceeds 0.1 v/deg; a higher-gain amplifier is required than if synchros are used as error sensing elements.

A.11.3 A-C Synchros

A pair of synchros is widely used as error sensing elements in a servo control system for accurately positioning a mechanical load. Basically, a synchro is a rotary device which is used to produce a correlation between an angular position and a voltage or set of voltages, so that the angular position of one shaft may follow that of another, even when remotely located without any mechanical linkage between them. They are widely used for instrumentation and data transmission, and in many similar applications requiring position co-ordination between the parts of complex servo control systems. Synchros are identified by trade names like *selsyn*, *Autosyn, Diehlsyn, Telesyn* by their manufacturers. There are several different types and applications of synchros, but in this section only the synchro transmitter, the synchro control transformer, and the synchro differential transmitter will be discussed.

A.11.4 Synchro Transmitter

A synchro transmitter has a Y or ∇-connected stator winding which resembles the stator of a three-phase induction motor. It can be shown that we need at least *three stator windings*

to uniquely indicate the angular position of a rotor by measuring the stator voltages. The number of stator windings are restricted to three for practical and economic reasons. The rotor is a salient-pole, dumbbell-shaped magnet with a single winding. A single-phase excitation voltage is applied to the rotor through two slip rings. The voltage may be 230 *v* at 50 Hz, 110 *v* at 400 Hz, or some other voltage and frequency depending upon the rating of the synchro. The schematic diagrams of a synchro transmitter are shown in Fig. A.25. The symbol "CX" is often used to designate a synchro transmitter. In synchro transmitter the rotor is mechanically positioned to transmit electrical information corresponding to angular position of the rotor with respect to the stator. The synchro transmitter is constructed to operate with synchro control transformers or synchro differential transmitters.

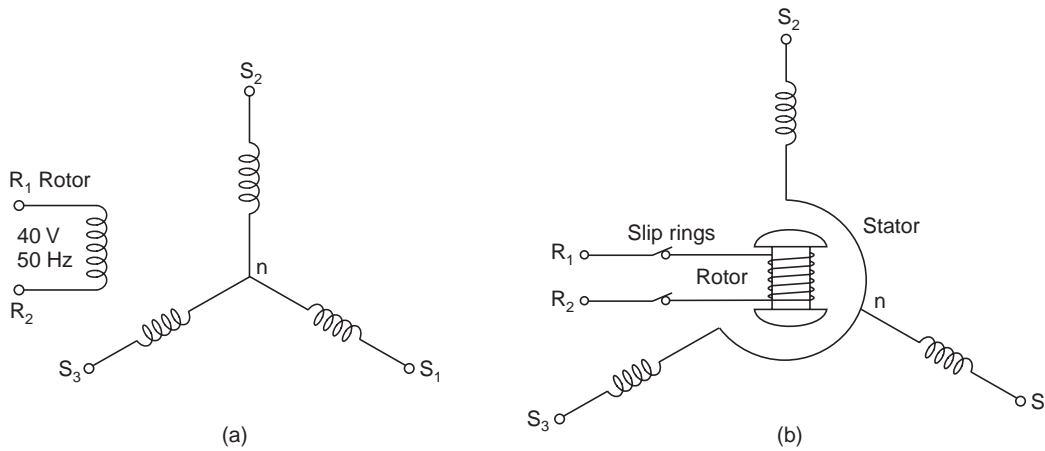



Fig. A-25 schematic diagram of synchro transmitter, CX

Let the AC voltage applied to the rotor be represented as

$$v_R(t) = V_R \sin \omega t \quad \text{(A.95)}$$

When the rotor is in the position as shown in Fig. A.25(*b*), the voltage induced across the stator winding S_2 and the neutral is maximum and is written as

$$v_{S2n}(t) = KV_R \sin \omega t \quad \text{(A.96)}$$

where K is a proportional constant. Taking this position of the rotor as the reference position, we note that it has to be rotated through 240° and 120° degrees in the counter clockwise (positive) direction to align it with S_1 and S_3, respectively. So, the voltages appearing across the terminals, S_1n and S_3n are given by

$$v_{S1n}(t) = KV_R \cos(-240°) \sin \omega t = -0.5KV_R \sin \omega t \quad \text{(A.97)}$$

$$v_{S3n}(t) = KV_R \cos(-120°) \sin \omega t = -0.5KV_R \sin \omega t \quad \text{(A.98)}$$

The three terminal voltages of the stator are

$$v_{S1S2} = v_{S1n} - v_{S2n} = -1.5KV_R \sin \omega t \quad \text{(A.99)}$$

$$v_{S2S3} = v_{S2n} - v_{S3n} = 1.5KV_R \sin \omega t \quad \text{(A.100)}$$

$$v_{S3S1} = v_{S3n} - v_{S1n} = 0 \quad \text{(A.101)}$$

The above equations show that, despite the similarity between the construction of a synchro stator and that of a three-phase induction machine, there are only single-phase voltages induced in the stator.

Now consider that the rotor is rotated through an angle θ in a counterclockwise direction, as shown in Fig. A.26.

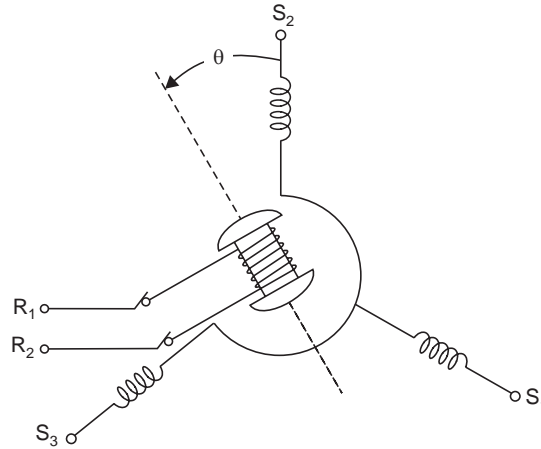



Fig. A-26 Rotor of synchro transmitter rotated through an angle θ from reference position

The voltages in each stator winding will vary as the function of the cosine of the rotor displacement θ; that is, the voltage magnitudes (omitting sin ω*t*) are

$$V_{S1n} = KV_R \cos(\theta - 240°) \tag{A.102}$$

$$V_{S2n} = KV_R \cos\theta \tag{A.103}$$

$$V_{S3n} = KV_R \cos(\theta - 120°) \tag{A.104}$$

The magnitudes of the stator terminal voltages become

$$V_{S1S2} = V_{S1n} + V_{nS2} = \sqrt{3}\, KV_R \sin(\theta - 120°) = \sqrt{3}\, KV_R \sin(\theta + 240°) \tag{A.105}$$

$$V_{S2S3} = V_{S2n} + V_{nS3} = \sqrt{3}\, KV_R \sin(\theta - 60°) = \sqrt{3}\, KV_R \sin(\theta + 120°) \tag{A.106}$$

$$V_{S3S1} = V_{S3n} + V_{nS1} = \sqrt{3}\, KV_R \sin\theta \tag{A.107}$$

A plot of these terminal voltages as the rotor shaft rotates from 0 degree to 360 degree is given in Fig. A-27. The variation of voltage with shaft-position imdicates that the synchro transmitter can be used to uniquely identify an angular position by measuring and identifying the set of voltages at the stator terminals.

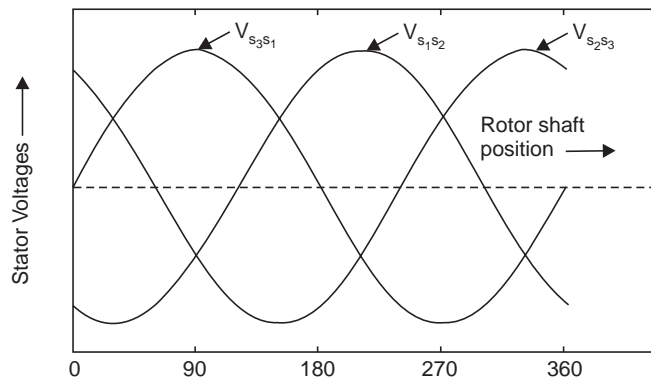



Fig. A-27 Variation of stator voltages with rotor angular position

A.11.5 Synchro Control Transformer

Since the function of an error detector is to convert the difference of two shaft positions into an electrical signal, a single synchro transmitter is apparently not adequate. A synchro control transformer is used in conjunction with a synchro transmitter to act as error sensor of mechanical components. Basically, the construction of a synchro control transformer is very similar to that of the synchro transmitter, except that the rotor is cylindrically shaped so that the air gap flux is uniformly distributed around the rotor (Fig. A-28). This is essential to a control transformer since its rotor terminals are usually connected to an amplifier. The cylindrical shape of the rotor of the synchro control transformer helps to keep the change of impedance in the rotor coil circuit with change in angular position. The symbol "CT" is often used to designate a synchro control transformer.

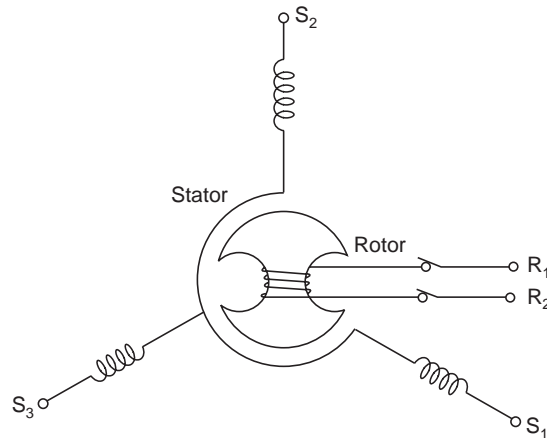



Fig. A-28 schematic diagram of synchro control transformer, CT

In order to use a pair of synchros as error detector in servo applications, the stator leads of the synchro transmitter is connected to the stator leads of a synchro control transformer as shown in Fig. A.29. The rotor of CX is connected to an ac source and the rotor coils of the synchro control transformer is connected to the input of an error amplifier. For small deviations, the voltage at the rotor terminals of the control transformer is proportional to the deviation between the two rotor positions (see Equation A-108 below).

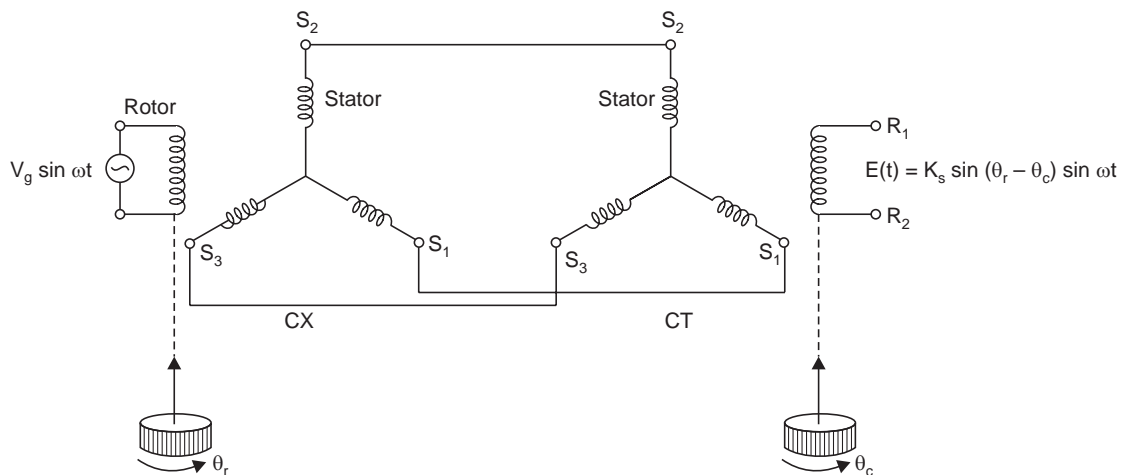



Fig. A-29 A pair of synchros used to detect angular discrepancy between θ_r and θ_c

With the arrangement in Fig. A-29, the voltages given by Equations (A.105), (A.106) and (A.107) are now impressed across the corresponding stator leads of the synchro control transformer. Because of the similarity in the magnetic construction, the flux patterns produced in the two synchros will be the same if all losses are neglected. For example, if the rotor coil of the transmitter is held parallel to the stator coil S_2, the direction of the flux in the air gap in the synchro control transformer will be such that the induced voltage in its rotor coil will be maximum when it is parallel to its stator coil S_2 and is minimum when the rotor coil makes an angle of 90° with the S_2 coil. With proper design, the minimum voltage can be made practically zero and this position of the rotor of the synchro control transformer, with respect to that of the control transformer (i.e., when the two rotors makes an angle of 90° with each other in space) is referred to as electrical zero position.

The shafts of the two synchros are considered to be in alignment when the control transformer voltage is zero. However, when the rotor position of the control transformer is rotated 180° from the electrical zero position, the terminal voltage is again zero. So, there are two null positions of the control transformer. So, it is found that when the two rotors are said to be in alignment, they are actually making an angle of 90° (or 270°) with each other in space. Therefore, if the control transformer rotor is rotated by an angle α from either of the null positions, the magnitude of the rotor voltage is proportional to sin α. The rotor voltage of a control transformer versus the difference in positions of the rotors of the transmitter and the control transformer is given in Fig. A.30.

For small angular displacement (approximately, up to 15 degree when $\sin \alpha \approx \alpha$) in the vicinity of the two null positions, the rotor voltage of the control transformer is approximately proportional to the difference between the positions of the rotors of the transmitter and the control transformer. For small discrepancies of the controlled and reference shaft positions, the transfer function of the synchro error detector can be written as

$$G(s) = \frac{E}{\theta_r - \theta_c} = \frac{E}{\theta_e} = K_s \tag{A.108}$$

where E = error voltage

θ_r = reference shaft position in degrees

θ_c = controlled shaft position in degrees

$\theta_e = \theta_r - \theta_c$ = error in angular positions between the reference shaft and the controlled shaft in degrees

K_s = sensitivity of the error detector in volts per degree.

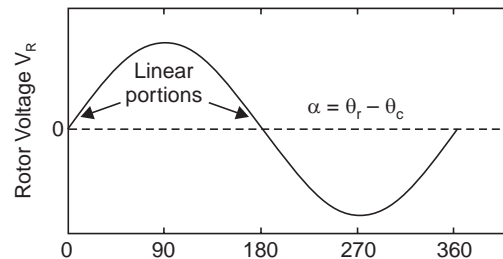



Fig. A-30 Variation of control transformer-rotor voltage with angular discrepancies of the rotors of control transmitter and control transformer.

A typical AC servo system employing a synchro error detector is shown in Fig. A.31. The purpose is to make the controlled shaft follow the angular variations of the reference input

shaft. The rotor of the control transformer is mechanically connected to the controlled shaft. The rotor of the synchro transmitter is connected to the reference input shaft. The error signal which appears at the rotor terminals of the control transformer is amplified which is eventually used to drive a two-phase induction motor.

Fig. A-31 Position control system of mechanical load using synchro transmitter and synchro control transformer as error sensing elements

When the controlled shaft is aligned with the reference shaft, the error voltage is zero and the motor does not turn. When an angular misalignment exists, an error voltage of relative polarity (phase) appears at the amplifier input. The motor will turn in such a direction as to correct this discrepancy.

From the characteristics shown in Fig. A.30, it is clear that K_s has opposite signs at the two null positions. However, in closed-loop systems, only one of the two null positions is the true null; the other one corresponds to an unstable operating point. Suppose that, in the system given in Fig. A.31, the synchros are operating close to the true null, and the controlled shaft lags behind the reference shaft; a positive error voltage will cause the motor to turn in the clockwise (say) direction to correct the lag. But if the synchros are operating close to the false null, for the same lag between θ_r and θ_c the error voltage is negative, which will cause the motor to turn in the anticlockwise direction to increase the lag. A larger lag in the controlled shaft will further increase the magnitude of the error voltage and cause the motor to turn further in the anticlockwise direction until K, is reversed in sign and the true null position is reached.

We can write the error voltage in Equation (A.108) explicitly showing its time-dependence as:

$$E(t) = K_s \theta_e(t) \sin \omega t \tag{A.109}$$

which has the form of an amplitude modulated signal with the frequency ω of the power supply acting as the carrier frequency and error in angular position $\theta_e(t)$ as the modulating signal. The variation of error voltage for a typical variation of $\theta_e(t)$ is shown in Fig. A.32.

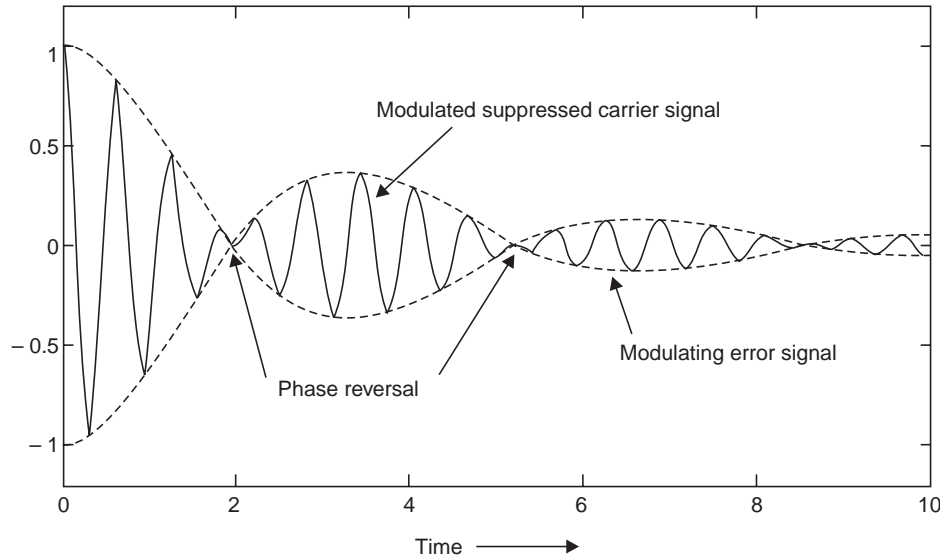



Fig. A-32 Amplitude modulated signal at the rotor of the synchro control transformer. The ac supply frequency ω is the carrier frequency

A.11.6 Synchro Differential Transmitter

A Synchro Differential Transmitter is a unit in which the rotor is mechanically positioned to modify electrical angular information received from a control transmitter or retransmit electrical information corresponding to the sum or difference, depending on the system winding. In a synchro differential transmitter, both the stator and the rotor have distributed windings similar to the winding on the stator of the synchro transmitter. In a position control system, it is often used to place a shaft which is the sum of two shaft angles. For this purpose, the unit is connected between a synchro transmitter and control transformer as shown Fig. A-33.

A.11.7 Resolvers

Resolvers (R) are induction-type devices used extensively for co-ordinate transformation and conversion from rectangular to polar coordinates. A resolver is essentially a variable transformer designed in such a way that its coupling coefficient between stator and rotor voltages vary as the sine or cosine of its rotor position. Usually there are two windings on the rotor and two on the stator at right angles to each other. The two rotor windings are connected to two voltage sources through 4 slip rings. Depending on the application, either the rotors or the stators may be used as the primary.

Fig. A-34 shows the rotors used as primary connected to sources $V_1 \sin \omega t$ and $V_2 \sin \omega t$ for which voltages induced in the stators are given by :

$$V_3 = K (V_1 \cos \theta + V_2 \sin \theta) \sin \omega t \quad \text{(A.110)}$$

$$V_4 = K (V_1 \sin \theta + V_2 \cos \theta) \sin \omega t, \text{ where K is the proportionality constant} \quad \text{(A.111)}$$

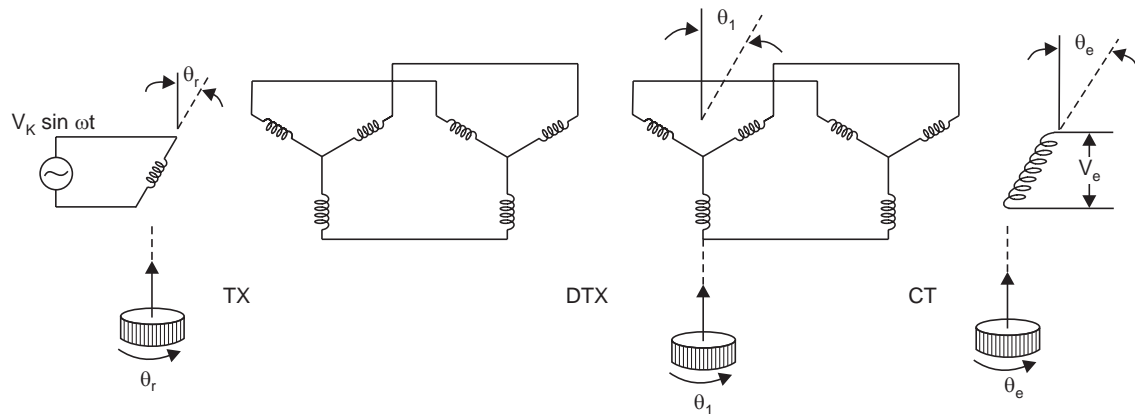



Fig. A-33 Use of synchro differential transmitter (DTX) to produce a voltage $V_e = K_s \sin(\theta_r + \theta_1 - \theta_e) \sin \omega t$

In the special case when supply voltage, $V_2 \sin \omega t$ to the second rotor coil is removed and coil terminals are short circuited, the stator voltages V_3 and V_4 become the resolved components of the voltage $V_1 \sin \omega t$ connected to the first coil of the rotor.

$$V_3 = K(V_1 \sin \omega t) \cos \theta \quad \text{(A.112)}$$

$$V_4 = K(V_1 \sin \omega t) \sin \theta \quad \text{(A.113)}$$

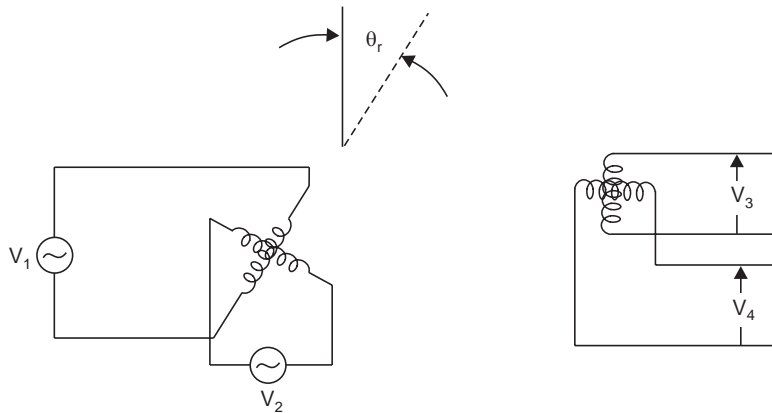



Fig. A-34 Resolver with 2 inputs and 2 outputs

A.11.8 Induction Potentiometers

An induction potentiometer or linear synchro transmitter, may be considered as a special form of resolver which provides accurate linear indication of shaft rotation about a reference position in the form of a polarized voltage whose magnitude is proportional to angular displacement, and whose phase relationship indicates direction of shaft rotation.

The principal difference between a resolver and an induction potentiometer is that the latter's output voltage varies directly as an angle and not as the sine of that angle. Its rotation is usually limited to less than ± 90°.

These devices are analogous to resistance potentiometers, but since they are induction-type components they have less restraining force acting upon their rotors, and hence are capable of providing better resolution. Naturally, induction potentiometers do not require sliders such

as those used in resistance types. Therefore, circuit noise caused by interruptions in slider contacts are eliminated, no wear and tear occurs due to mechanical frictions of moving parts, and accuracy is consequently continuously maintained at the original level throughout the operational life of these types of components. These advantageous features are virtually a necessity in certain applications. For example, these devices are widely used in gyroscope systems where low restraining torques and low pickoff angular errors are required.

In general, induction potentiometers find use in applications where resistive potentiometers are impractical, principally because of the following characteristics :

(*i*) induction potentiometers, having no wiper contacts, may be used as gyroscope pickoffs since they contribute less spurious friction torque;

(*ii*) input and output are isolated;

(*iii*) resolution is infinite;

(*iv*) noise level is low; and

(*v*) the total angle of travel is limited to less than 180°.

Constructed very much like resolvers, induction potentiometers differ in that their windings and slots are not uniformly distributed, but instead are deliberately modified to produce a linear output. Normally they have only one excitation winding and one output winding. The former is usually carried on the rotor and connected by means of two slip rings and two brushes. Accuracies on the order of 0.1 percent are obtainable.

A.12 THE LINEAR VARIABLE DIFFERENTIAL TRANSFORMER (LVDT)

The linear variable differential transformer (LVDT) is a special transformer which works based on the principle of mutual inductance. The LVDT produces an electrical signal that is proportional to the linear displacement of a movable armature or core so that it can be used as a transducer for measuring linear mechanical displacement by measuring its electrical output.

The LVDT has a cadmium-plated cylindrical housing with a hollow center through which an armature, or piece of magnetic material can move without touching the body of the cylinder. The differential transformer has one primary and two secondary windings in series opposition (vide Fig. A.35). The form of coil is cylindrical shaped which remains stationary and is enclosed in a protective magnetic shield. With negligible friction, the differential transformer gives practically infinite life and resolution for measuring straight-line displacement.

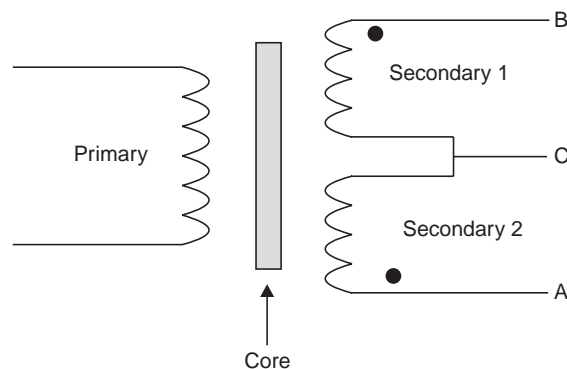



Fig. A-35 LVDT coil windings along with movable core

When the primary is energized by an ac supply, voltage is induced in the secondary through the core by transformer action. The position of the core, determines the magnitude of the induced voltage to the two secondary coils having the same number of turns. If the core is precisely positioned at the center, the two secondary voltages will be equal and of opposite polarity when measured with respect to common terminal C (see Fig. A-35, the dots represent the start of winding direction). So the voltage measured between terminals AB will be zero. This position of the core is referred to as null or zero position. Now if the core is moved upward through a displacement of **x** units from this zero position, the magnitude of voltage induced in the coil CB will be more than that of the voltage induced on the coil CA. So the voltage between the terminals AB will be non-zero and its amplitude will be proportional to the displacement **x**. Similarly if the core is moved downward through the same displacement **x**, the voltage between the terminals AB will be the same in magnitude but opposite in phase. Therefore, by measuring the magnitude and phase one can measure the displacement and its direction of displacement from the null position.

This characteristic of the LVDT, vide Fig. A-36, makes it a useful transducer to translate a linear mechanical displacement, such as lever position, roller position, mechanical feed, cam position, etc., into an exactly proportional electric signal for local or remote transmission for measurement and control. The armature of the transformer is mounted on the spring-loaded plunger that is placed against the device to be measured. Thus, any movement of the transducer tip and armature provides a direct proportional electrical signal for indication, recording, or control.

The transducer consists of a hardened adjustable contact tip, in a locking assembly, mounted on a spring-loaded shaft. The shaft travels through and is held in a linear position by a bearing. The core is attached to the other end of the plunger shaft. The transformer coil is held in position by set of screws through the transducer housing.

Using a phase sensitive rectifier (vide Fig. A-37), the LVDT will provide a dc signal proportional to displacement. Using solid state rectifiers connected to the output legs of the secondary, the dimension of LVDT does not change. The positive or negative polarity of the output shows whether the armature is nearer one end of the LVDT or the other.

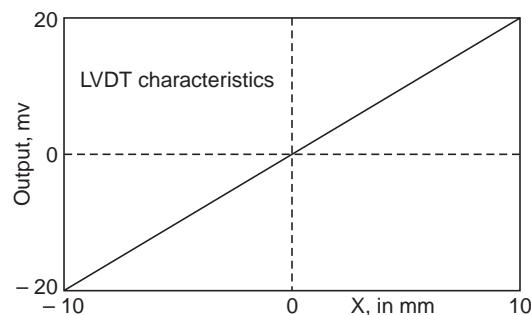



Fig. A-36 Input-output characteristic of LVDT. In case of ac outputs, the positive and negative signs indicate in-phase and out of phase signals, whereas for dc outputs they represent signals of opposite polarity

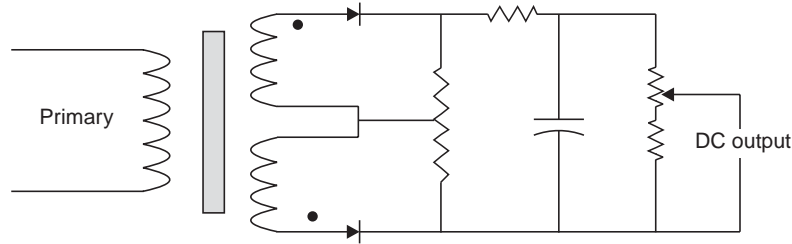



Fig. A-37 LVDT with DC output

Advantages of LVDT as linear displacement transducer

The LVDT has distinct advantages over other devices used as a transducer for mechanical positions. Some of these advantages are :

1. In absence of any mechanical contact between the primary and secondary windings and the armature, there is no friction or hysteresis.
2. There is no mechanical wear and tear ensuring virtually infinite life.
3. Linear output assures accurate measurement with direct indicating instruments.
4. The resolution is very high, limited only by the indicating and control equipment.
5. The electrical output is completely isolated from the input, allowing addition or subtraction of signals without buffer amplifiers.
6. High-level output simplifies electronic circuitry.
7. Over-ranging does not cause any damage or permanent change in characteristics.
8. The LVDT is rugged and shock-resistant, and virtually maintenance-free.

A. Linearity of LVDT: Absolute linearity is defined as the largest deviation (or error) voltage of transducer output from the theoretical ideal, measured between zero and full-scale displacement and expressed as a percent of the full-scale output.

B. LVDT Sensitivity: At high-frequency excitation, LVDT sensitivity is practically independent of changes in ambient temperature. Sensitivity ratings are therefore measured at constant voltage, and expressed as millivolts output per volt of excitation per unit displacement (usually 0.01 mm) (mV/V/0.01 mm).

At low-frequency (5OHz) excitation, the voltage sensitivity of LVDTs changes with temperature. It is therefore normally expressed at constant current (V/A/mm), which may be converted to mV/V/.01 mm as follows,

$$K_s = \frac{\text{V/A/mm}}{\text{primary impedance}}$$

A.13 HYDRAULIC SERVOMOTOR

The schematic diagram of a hydraulic servomotor is shown in Fig. A.38, which is basically a pilot-valve-controlled hydraulic power amplifier and actuator. The positioning of pilot valve needs very little power but it can control a very large power output to drive heavy mechanical loads.

The opening of the ports **a** and **b** shown in Fig. A.38 are normally made wider than the corresponding valves V_1 and V_2. For this reason, there is always leakage through the valve. This helps to improve both the sensitivity as well as the linearity of the hydraulic servomotor.

If the pilot valve is so positioned that the coverage of ports **a** and **b** by the respective valves V_1 and V_2 are equal, the leakage fluid-flow through the ports **a** and **b** will also be equal, making the pressures on two sides of the power piston equal (i.e., $P_1 = P_2$). This position of the pilot valve is taken as the zero or neutral position ($x = 0$). Now if the pilot valve is slightly moved to the right, $x > 0$, then the exposure of the port **b** to the source pressure is increased, at the same time the exposure of the port **a** to the sink is also increased, making $P_2 > P_1$. The force developed due to the pressure differential will move the power piston, together with any load attached to it, to the left. Similarly, if the pilot valve is moved to the left from the neutral position, $x < 0$, the pressure on left hand side of the power piston will be more than that in the right hand side ($P_1 > P_2$), as a result, the power piston, along with the attached load, will move to the right.

In the analysis below, it is assumed that the hydraulic fluid is incompressible. In order to derive the mathematical model of the hydraulic motor we introduce the following variables:

Q = rate of flow of working fluid (oil or any suitable liquid) to the power cylinder

$\Delta P = P_1 - P_2$ = pressure difference across the power piston

x = displacement of pilot valve

The flow rate Q is a nonlinear function of the variables x and ΔP and can be written as:

$$Q = f(x, \Delta P)$$

In order to obtain a transfer function model, we can linearize the above nonlinear equation around the normal operating point Q_s , X_s, ΔP_s as shown below:

$$Q - Q_s = \frac{\delta f}{\delta x}(x - x_s) + \frac{\delta f}{\delta \Delta P}(\delta \Delta P - \delta \Delta P_s) \qquad (A.114)$$

At the quiescent point $Q_s = f(x_s, \Delta P_s)$, the partial derivatives evaluated at $X = X_s$, $\Delta P = \Delta P_s$ may be taken as constants and designated as :

$$k_1 = \left.\frac{\partial f}{\partial x}\right| (X = X_s, \Delta P = \Delta P_s > 0$$

$$k_2 = \left.\frac{\partial f}{\partial \Delta P}\right| (X = X_s\ , \Delta P = \Delta P_s) > 0$$

Using incremental variables $q = Q - Q_s$, $x = X - X_s$, and $\Delta p = \Delta P - \Delta P_s$, Equation (A.114) may, therefore, be written as

$$q = k_1 x - k_2 \Delta p \qquad (A.115)$$

Fig. A.39 depicts the linearized characteristic curves of the hydraulic servomotor with the pilot valve displacement x as a parameter. The characteristic curves are equispaced straight lines with equal increments of the parameter x. The characteristic curve passes through the origin with negative slope for $x = 0$. The linearized characteristic will enable us to derive the transfer function model of the hydraulic servo motor.

With reference to Fig. A.38, the mass balance equation may be written as

$$A\rho dy = q dt$$

where A is the cross-section of the power cylinder, ρ is the density of the fluid and dy is the displacement of the power cylinder.

The velocity of the load can be found from the above equation as:

$$dy/dt = q/A\rho$$

So for a faster response of the load, the cross-section of the cylinder, A should be small.

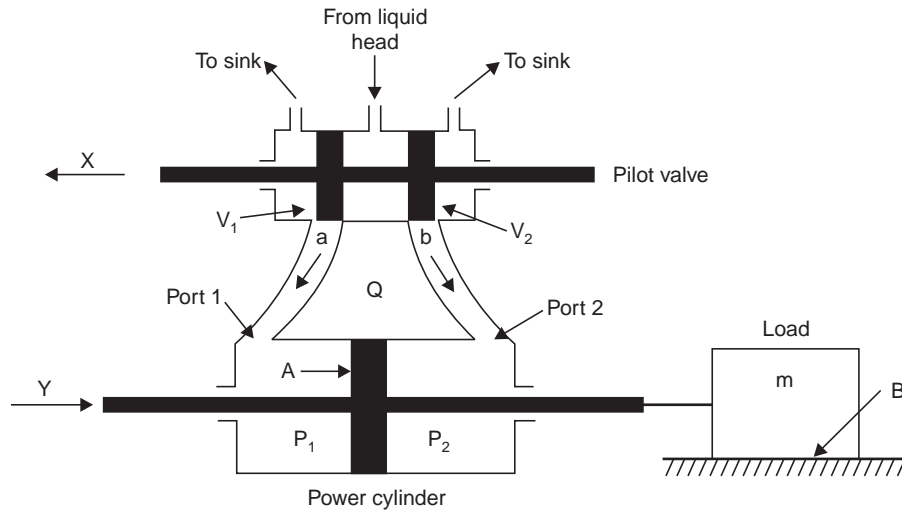



Fig. A.38 Hydraulic motor (actuator)

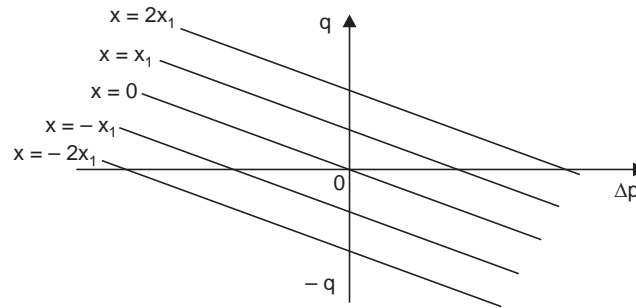



Fig A.39 Linearized characteristic of hydraulic motor

We can write Equation (A.115) as:

$$\Delta p = \frac{1}{k_2}\left(k_1 x - \mathrm{A}\rho\frac{dy}{dt}\right)$$

The force developed by the power piston is equal to the product of the piston area A and the pressure difference Δp.

That is, the force developed by the piston = AΔp

or,

$$\mathrm{A}\Delta p = \frac{\mathrm{A}}{k_2}\left(k_1 x - \mathrm{A}\rho\frac{dy}{dt}\right)$$

If the pressure difference Δp is very high, the area of the piston A can be made small for a prescribed force needed for a given mechanical load. Besides, the volume of oil in the cylinder along with the size and weight of the hydraulic motor can be made small by making the fluid supply pressure sufficiently high.

Consider the case that the power piston is used to move a mechanical load consisting of a mass m with viscous friction coefficient B. Then the force developed by the power piston is utilized to accelerate the load and overcome the viscous friction. So we can write

$$m\ddot{y} + \mathrm{B}\,\dot{y} = \frac{\mathrm{A}}{k_2}\left(k_1 x - \mathrm{A}\rho\dot{y}\right) \quad \text{or} \quad m\ddot{y} + \left(\mathrm{B} + \frac{\mathrm{A}^2\rho}{k_2}\right)\dot{y} = \frac{\mathrm{A}\,k_1}{k_2}x \tag{A.116}$$

where m is the mass of the load and B is the coefficient of viscous-friction and the inertia of the pistons are negligible.

With the pilot valve displacement x as the input and the power piston displacement y as the output, we find from Equation (A.116), that the transfer function for the hydraulic servomotor is given by :

$$\frac{Y(s)}{X(s)} = \frac{1}{s\left[\left(\frac{mk_2}{Ak_1}\right)s + \frac{Bk_2}{Ak_1} + \frac{A\rho}{k_1}\right]} = \frac{k}{s(\tau s + 1)} \tag{A.117}$$

where,

$$k = \frac{1}{\frac{Bk_2}{Ak_1} + \frac{A\rho}{k_1}} \quad \text{and} \quad \tau = \frac{mk_2}{Bk_2 + A^2\rho}$$

So, the transfer function of the hydraulic motor is a second order one as reveated by. Equation (A.117). If the time constant $\tau = mk_2/(Bk_2 + A^2\rho)$ is negligibly small. The transfer function can be simplified as a first order integrator :

$$\{Y(s)/X(s)\} = k/s$$

A detailed analysis considering oil leakage, compressibility (including the effects of dissolved air), expansion of pipelines, and the likes, the transfer function may be shown to be of the form :

$$\frac{Y(s)}{X(s)} = \frac{k}{s(s\tau_1 + 1)(s\tau_2 + 1)}$$

where τ_1 and τ_2 are time constants. The time constants depend on the volume of oil in the operating system and time constants decreases with decreasing fluid volume.

The state variable representation of the dynamics in Equation (A.116) can be readily obtained with the following choice of states

$$y(t) = x_1(t),\ \dot{y}(t) = x_2(t) \ \text{ and } \ x(t) = u(t)$$

The state equation will be similar to Equation (A.50) with $a_1 = 0$, $a_2 = (Bk_2 + A^2\rho)/mk_2$ and $b_1 = Ak_1/mk_2$.

A.14 PNEUMATIC CONTROLLER MECHANISMS

Baffle Nozzle The basic element in the motion-balance type of pneumatic mechanism is the baffle-nozzle system shown in Fig. A.40. As in the case of electronic amplifier, a potential source is needed to provide the proper working condition and source of power to the pneumatic amplifier. Compressed air at 20 psi (pounds per square inch) is normally used as the source of power for pneumatic amplifiers. With reference to Fig. A.40*a*, if the baffle is pressed against the nozzle ($\delta = 0$), the pressure inside the nozzle is equal to the supply pressure p. As the baffle is pulled away from the nozzle, the pressure in the nozzle decreases. A typical curve relating nozzle pressure to baffle-nozzle displacement is shown in Fig. A.40(*b*). The characteristic shows that the pressure is reduced to 0 when the baffle has moved only a few thousandths of an inch, which means the gain of the baffle, $m = |\, dp/d\delta \,|$, is very high (similar to the case of an open loop operational amplifier in electronics). For this reason a baffle nozzle is rarely used in an open loop configuration. The restriction R_1 should be small compared to that of the nozzle for its proper functioning.

A.14.1 Pneumatic Proportional Controller

The baffle-nozzle system of the Fig. A.40 is too sensitive to be used without feedback in a proportional controller. A spring-loaded bellows is used to provide the feedback around the

baffle nozzle, as shown in Fig. A-41. The right end of the bellows is fixed, and the left end is free to move horizontally as the pressure in the bellows changes. The upper end of the baffle is attached to the bellows which will be acting as a movable hinge and moves horizontally as the bellows expands or contracts. At low frequencies the pressure in the bellows is equal to that in the nozzle at any instant. In Fig. A-41, e may represent the error between the set point and the position of a mechanical load and the displacement, x of the lower end of the baffle is proportional to the error (*i.e.,* $x = Ce$ where C is a constant)

Assume that the system is in steady-state conditions. Now if a step change in error is introduced, the lower end of the baffle moves to the right by an amount $\Delta x = C\Delta e$. This decreases the separation between baffle and nozzle resulting in an increase of pressure inside the nozzle according to the characteristics in Fig. A.40(*b*). The increase in pressure in the nozzle is transmitted to the bellows, as a result, the bellows will expand and move the upper end of the baffle to the left. This motion to the left is opposite to that of the lower end of the baffle; thereby nullifying partly the effect of the original motion caused by the step change in error. As a result of this feedback, the change in pressure, p will be much reduced compared to that of the open loop case.

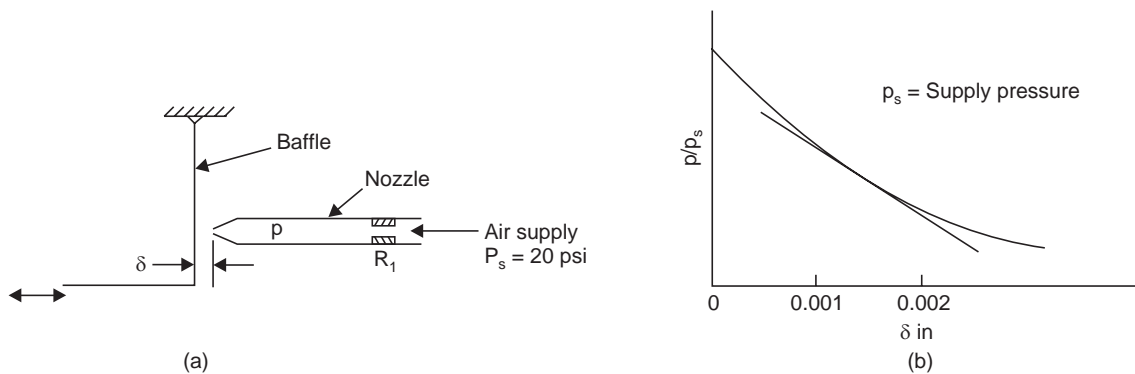



Fig A.40 Baffle-nozzle system and its characteristic

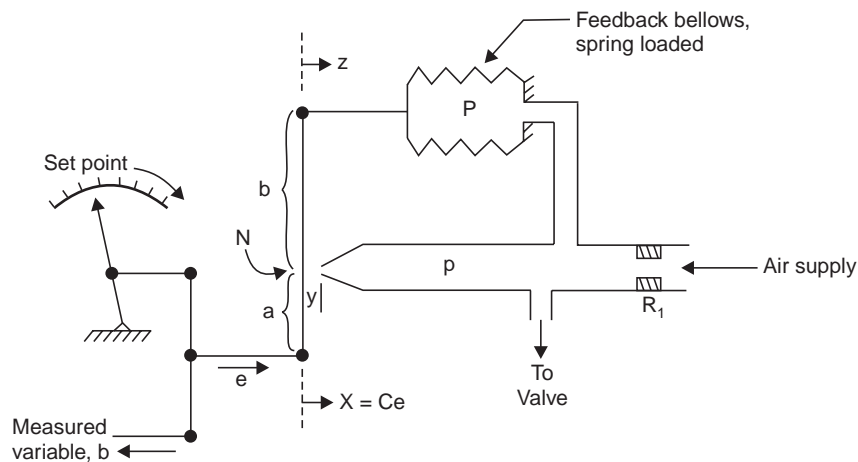



Fig. A.41 Proportional controller mechanism

In Fig. A.41, y represent the perpendicular distance of the baffle from the nozzle orifice (N is the foot of the perpendicular). Let x, y, z be the incremental displacements of the baffle

measured from their steady state values when the pressure inside the nozzle is p_s. In particular let $x = 0$, $z = 0$ and $y = 0$ when incremental pressure $P = p - p_s = 0$. If the baffle nozzle operates in the linear range of its characteristic curve, the pressure inside the nozzle may be written as a functional of y

$$p = P - P_s = my \tag{A.118}$$

where m is the slope of the linear portion of the curve of Fig. A.40*b*.

The motion involved in an actual mechanism is very small and the baffle remains practically vertical throughout its range of motion and y is so small over the range of operation that the baffle is essentially pivoted at the point N.

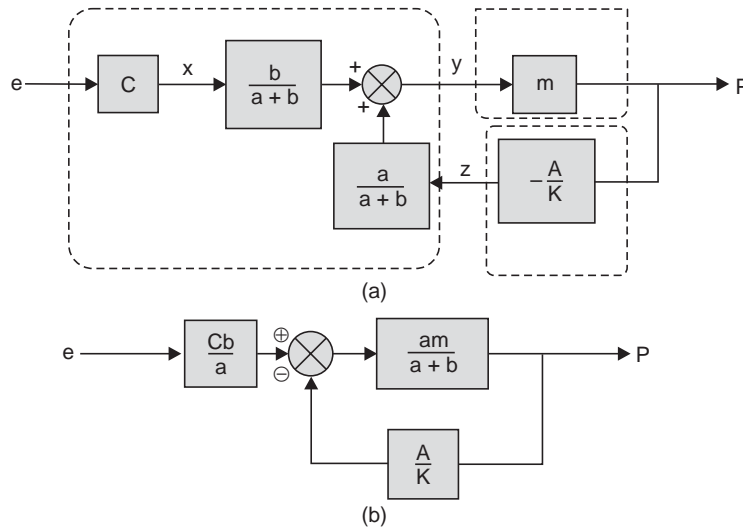



Fig. A-42 Block diagram for proportional controller

By using properties of similar triangles and with z held to a constant value, y is related to x as

$$y = bx/(a + b)$$

whereas with x = constant, y is related to z as

$$y = az/(a + b)$$

Therefore, when both x and z vary, y is found to be

$$y = \frac{b}{a+b}x + \frac{a}{a+b}z = \frac{b}{a+b}Ce + \frac{a}{a+b}z; \quad \text{where } x = Ce \tag{A.119}$$

Since the displacement z is against a spring with constant K, we can write

$$Kz = -A(P - P_s) = -Ap \tag{A.120}$$

where A is the surface area of the bellows on which the pressure p acts. The negative sign in Equation (A.120) has been included to take into account the fact that an increase in p will cause z to decrease as the upper end of the baffle moves to the left.

From Equations (A.118) and (A.119) we have

$$p = \frac{mb}{a+b}Ce + \frac{ma}{a+b}z \tag{A.121}$$

Eliminating z from the Equations (A.120) and (A.121) gives

$$p = \frac{mb}{a+b}\mathrm{C}e - \frac{ma}{a+b}\frac{\mathrm{A}}{\mathrm{K}}p$$

Hence, rearranging this last equation gives

$$\frac{p}{e} = \frac{\dfrac{\mathrm{C}mb}{a+b}}{1 + \dfrac{ma\mathrm{A}}{(a+b)\mathrm{K}}} \tag{A.122}$$

If $ma\mathrm{A}/(a + b)\mathrm{K} >> 1$, Equation (A.122) can be approximated by

$$\frac{p}{e} = \mathrm{C}\frac{b}{a}\frac{\mathrm{K}}{\mathrm{A}} \tag{A.123}$$

From relation (A.123) it is obvious that the pressure change inside the nozzle is proportional to the error e in mechanical displacement.

Equations (A.118), (A.119) and (A.120) may also be represented in a block diagram form as shown in Fig. A-42(*a*). The Fig. A.42(*b*) is an equivalent form of Fig. A-42(*a*) and it clearly reveals the presence of negative feedback. The relation (A.123) could have been deduced from Fig. A.42(*b*).

A.14.2 Pneumatic Relay

The pressure signal from the controller mechanisms of Fig. A.41 cannot be used directly to operate a large control valve which needs a large air flow. A power amplifier is needed for acting on a large control valve. A pneumatic relay is such a power amplifier. The relay consists of bellows A of small volume and a specially designed valve B (Fig. A.43). The variation of pressure inside the nozzle is transmitted to the bellows A and as the pressure increases, the bellows expands pushing the ball down the cavity of the valve B, thereby decreasing the leakage to the atmosphere and increasing the pressure p_v. The relay is made in such a way that the output pressure p_v, is proportional to the signal pressure p. The flow rate of air from the relay will be much greater than the flow rate through the nozzle because the resistance to the air supply to the relay can be made very small compared to the resistance R_r in the nozzle. A relay valve can be designed as direct-acting, in which case the output pressure is directly proportional to input pressure, or it can be inverse-acting, in which case the output pressure is inversely proportional to input pressure. The relay shown in Fig. A-43 is direct-acting.

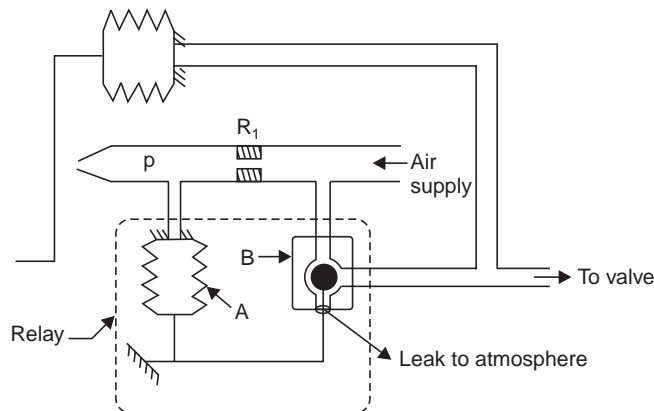



Fig. A.43 Pneumatic proportional controller with relay

A.14.3 Pneumatic PID Controller

A three term proportional-integral-derivative-controller (PID) is shown in Fig. A.44. The right end of the right bellows and the left end of the left bellows are fixed and the upper end of the flapper will be displaced in response to the differences in pressures p_1 and p_2 in the two bellows.

For analysis of the controller shown in Fig. A.44 a steady state condition is assumed when the pressures in both the bellows along with that in the nozzle are equal to some reference pressure p_s. The values of x, y and z are taken to be zero at steady state. The position of the upper end of the baffle will be related to the pressures in the bellows as

$$Kz = -A(p_1 - p_s) + A(p_2 - p_s) \quad (A.124)$$

where A is the surface area common to both the bellows against which the pressures act and K is the spring constant for the pair of bellows.

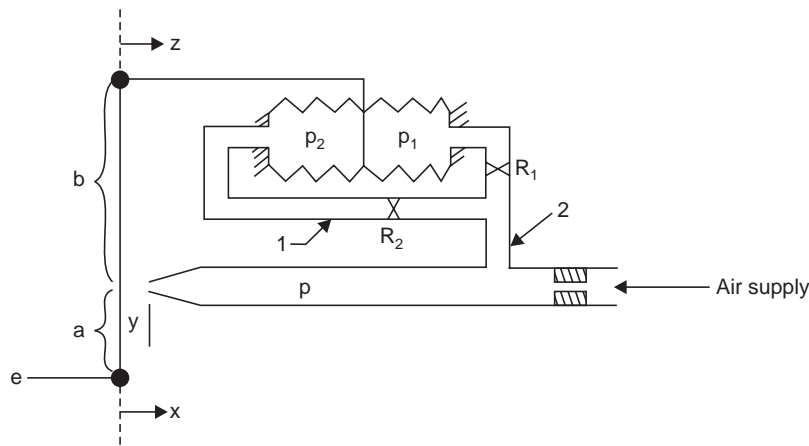



Fig. A.44 Parallel realisation of pneumatic PID controller. For series realisation R_1 should be placed at position 2 in the figure.

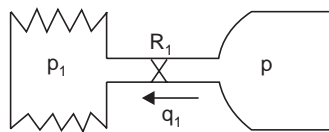



Fig. A.45 A single bellows

In order to obtain the mass balance equations consider the right-hand bellows as a separate system, as shown in Fig. A.45. Let the ambient temperature be constant at T and the volume of the bellows be constant at V, which is true for small displacement of the upper end of the flapper. The mass balance equation around the bellows shown in Fig. A.45 is given by

$$q_1 = dM/dt \quad (A.125)$$

where q_1 = flow rate into bellows, moles/sec

M = air in bellows, moles

Assuming ideal gas law, we have

$$M = \frac{P_1 V}{RT} \quad (A.126)$$

Combining Equations (A.125) and (A.126) yields

$$q_1 = C_1 \frac{dP_1}{dt} \quad \text{where } C_1 = \frac{V}{RT} \tag{A.127}$$

The term C_1 is called the capacity of the bellows. Since the mass flow rate is proportional to the pressure drop across the resistance, we have

$$q_1 = (P - P_1)/R_1 \tag{A.128}$$

Combining Equations (A.127) and (A.128) yields $\tau_1 \frac{dP_1}{dt} = P - P_1$ (A.129)

where $\tau_1 = R_1C_1 = R_1V/RT$ is the time constant of the right-hand bellows.

In a similar manner, the following differential equation can be written for the left-hand bellows of Fig. A.44.

$$\tau_2 \frac{dP_2}{dt} = P - P_2 \tag{A.130}$$

Combining Equations (A.130), (A.129), (A.124), (A.118) and (A.119) we can obtain the desired relationship between the pressure output p and mechanical displacement e.

Let us use the following deviation variables in the above equations:

$$p = P - P_s$$
$$p_1 = P_1 - P_s$$
$$p_2 = P_2 - P_s$$

Now representing the Laplace transform of $p(t)$ as $P(s)$ etc., we can write the above relations in terms of transformed variables as:

$$P(s) = mY(s) \tag{A.131}$$

$$Y(s) = \frac{bC}{a+b} E(s) + \frac{a}{a+b} Z(s) \tag{A.132}$$

$$Z(s) = \frac{-A}{K} P_1(s) + \frac{A}{K} P_2(s) \tag{A.133}$$

$$\frac{P_1(s)}{P(s)} = \frac{1}{\tau_1 s + 1} \tag{A.134}$$

$$\frac{P_2(s)}{P(s)} = \frac{1}{\tau_2 s + 1} \tag{A.135}$$

Solving Equations (A.131) through (A.135) simultaneously yield the following transfer function:

$$\frac{P(s)}{E(s)} = \frac{\dfrac{bC}{(a+b)}}{\dfrac{a}{a+b}\dfrac{A}{K}\left(\dfrac{1}{\tau_1 s + 1} - \dfrac{1}{\tau_2 s + 1}\right) + \dfrac{1}{m}} \tag{A.136}$$

If the slope m of baffle-nozzle characteristic is very large, the term $1/m$ can be neglected in Equation (A.136) and the equation can now be placed in the standard form of PID controller transfer function.

$$\frac{P(s)}{E(s)} = \frac{bKC}{aA}\left(\frac{\tau_1 + \tau_2}{\tau_2 - \tau_1}\right)\left[1 + \left(\frac{\tau_1\tau_2}{\tau_1 + \tau_2}\right)s + \frac{1}{(\tau_1 + \tau_2)s}\right] \tag{A.137}$$

It is of the form : $\frac{P(s)}{E(s)} = \left[K_p + K_d s + \frac{K_I}{s}\right]$

It is apparent from Equation (A.137) that the proportional gain K_p can be varied by varying the ratio b/a and the derivative constant K_d , integral constant K_I, can be changed by changing the resistances R_1 and R_2.

A.14.4 PI Controller

The three-mode controller becomes a PI controller by simply removing resistance R_1, that is, make $\tau_1 = 0$, For this case, Equation (A.137) reduces to

$$\frac{P(s)}{E(s)} = \frac{b\,KC}{a\,A}\left[1 + \frac{1}{\tau_2 s}\right] \tag{A.138}$$

A.14.5 PD Controller

If the air supply path at R_2 is closed such that $\tau_2 \to \infty$, the three term controller will be reduced to a PD controller. Under this condition, Equation (A.137) becomes

$$\frac{P(s)}{E(s)} = \frac{b\,KC}{a\,A}[1 + \tau_1 s] \tag{A.139}$$

The state variable representation of the pneumatic controller can be easily obtained from Equations (A.130), (A.129), (A.124), (A.119) and (A.118).

A.15 INVERTED PENDULUM

A free body diagram of an inverted pendulum mounted on a motor-driven cart is shown in Fig. A.46. It represents a model of the attitude control problem of a space booster on take-off. The space booster needs to be maintained in a vertical position but left to itself it will fall over in any direction since the vertical position is an unstable one. The purpose of the attitude control problem is to keep the space booster in the vertical position by applying an external control force u to keep it vertical. We shall derive a mathematical model of the vertical pendulum by considering that the pendulum can move only one plane, the plane of the paper.

We define the following variables :

M = mass of the cart
m = mass of the pendulum
$2l$ = length of the pendulum
$\theta = \theta(t)$, the angle the rod makes with the vertical line
$u(t)$ = the control force applied to the cart
$H(t)$ = the horizontal component of force
$V(t)$ = the vertical component of force
μ_c = the cart friction coefficient
$f_c(t) = \mu_c$ sign $(\dot{x})$, the frictional force at the cart wheels on the track assuming the wheels do not slip

Let (x, y) represent the coordinate of the point P on the cart at which the rod is supported and (x_g, y_g) be the coordinate of the center of gravity for the pendulum mass located at its geometric center. Therefore, we have

$$x_g = x + 1 \sin \theta \tag{A.140}$$

$$y_g = 1 \cos \theta \tag{A.141}$$

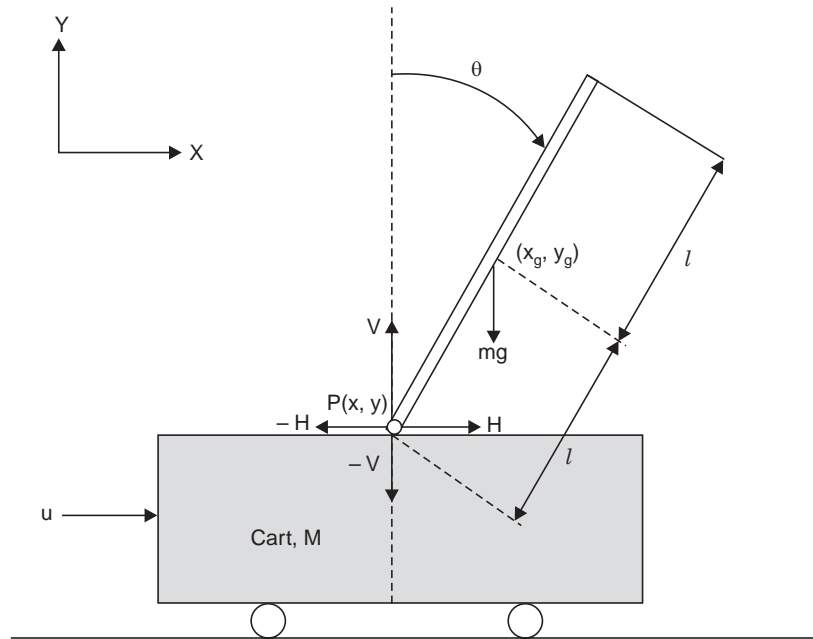



Fig. A-46 Inverted pendulum

We are now ready to write down the equations modeling the systems. The equation that described the rotational motion of the stick about its center of gravity (COG) is obtained by applying the rotational version of Newton's second law. Summing the moments about the COG of the stick, we obtain

$$J\frac{d^2\theta}{dt^2} = V\,l\sin\theta - H\,l\cos\theta \tag{A.142}$$

where
$$J = \int_0^1 mr^2 dr = \frac{ml^2}{3} \tag{A.143}$$

is the moment of inertia of the stick with reference to its COG. Next, we write the equation that described the horizontal motion of the COG of the stick. Applying Newton's second law along the x-axis yields

$$m\frac{d^2}{dt^2}(x + l\sin\theta) = H \tag{A.144}$$

Performing the required differentiation yields

$$H = m\frac{d}{dt}[\dot{x} + l\cos\theta\,\dot{\theta}] = m[\ddot{x} + l(-\dot{\theta}^2\sin\theta + \ddot{\theta}\cos\theta)]$$

or,
$$H = m\ddot{x} - ml\dot{\theta}^2\sin\theta + ml\ddot{\theta}\cos\theta \tag{A.145}$$

The equation that describes the vertical motion of the COG of the stick is also obtained by Newton's second law:

$$m\frac{d^2}{dt^2}(l\cos\theta) = V - mg \tag{A.146}$$

Performing the differentiation

$$V - mg = m\frac{d}{dt}[-l\sin\theta\dot{\theta}] = ml(-\dot{\theta}^2\cos\theta - \ddot{\theta}\sin\theta)$$

or
$$V = mg - ml\dot{\theta}^2\cos\theta - ml\ddot{\theta}\sin\theta \tag{A.147}$$

Finally, we apply Newton's second law to the cart to get

$$M\frac{d^2x}{dt^2} = u - H - f_c \tag{A.148}$$

Using the value of H from (A.145) we get,

$$u - f_c = H + M\ddot{x} = m\ddot{x} - ml\,\dot{\theta}^2 \sin\theta + ml\ddot{\theta}\cos\theta + M\ddot{x} \tag{A.149}$$

with $a = \dfrac{1}{M+m}$, we can simplify equation (A.149) as

$$\ddot{x} = -\,mal\ddot{\theta}\cos\theta + mal\dot{\theta}^2 \sin\theta - af_c + au \tag{A.150}$$

Now, substituting the values of H and V from (A.145) and (A.147)) into Equation (A.142), we get

$$J\ddot{\theta} = (mg - ml\dot{\theta}^2\cos\theta - ml\ddot{\theta}\sin\theta)\, l\sin\theta - (m\ddot{x} - ml\,\dot{\theta}^2\sin\theta + ml\ddot{\theta}\cos\theta)\, l\cos\theta \tag{A.151}$$

After simplification we get, $\quad J\ddot{\theta} = mgl\sin\theta - ml^2\ddot{\theta} - m\ddot{x}l\cos\theta \qquad$ (A.152)

We can substitute (A.150) into (A.152) to obtain $\ddot{\theta}$ as follows:

$$\ddot{\theta} = \frac{mgl\sin\theta - m^2l^2a\dot{\theta}^2\dfrac{\sin 2\theta}{2} - mla\cos\theta\,(u - f_c)}{J - m^2l^2\, a\cos^2\theta + ml^2} \tag{A.153}$$

We can choose states $x_1 = \theta$ and $x_2 = \dot{\theta}$, and using relation (A.143), we can write the state Equation from (A.153) as:

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{g\sin x_1 - mla\, x_2^2 \sin\dfrac{(2x_1)}{2}}{\dfrac{4l}{3} - mla\cos^2 x_1} - \frac{a\cos x_1\,(u - f_c)}{\dfrac{4l}{3} - mla\cos^2 x_1} \tag{A.154}$$

We next substitute $\ddot{\theta}$ obtained from (A.151) into (A.150) to get

$$\ddot{x} = -\,mla\cos\theta\left(\frac{mgl\sin\theta - m\ddot{x}l\cos\theta}{J + ml^2}\right) + mla\,\dot{\theta}^2\sin\theta + (u - f_c)a$$

$$= -\frac{3}{4}\frac{mla\cos\theta}{ml^2}(mgl\sin\theta - m\ddot{x}l\cos\theta) + mla\,\dot{\theta}^2\sin\theta + (u - f_c)a$$

$$= -\frac{3}{4}mga\sin\theta\cos\theta + \frac{3}{4}ma\,\ddot{x}\cos^2\theta + mla\,\dot{\theta}^2\sin\theta + (u - f_c)a$$

Collecting coefficients of $\ddot{x}$ and simplification we get,

$$\ddot{x} = \frac{-\,mga\dfrac{\sin(2\theta)}{2} + \dfrac{4mla}{3}\dot{\theta}^2\sin\theta + \dfrac{4a}{3}(u - f_c)}{\left(\dfrac{4}{3} - ma\cos^2\theta\right)} \tag{A.155}$$

Now when frictional force f_c is negligible, Equation (A.155) can be simplified, after substituting the value of a, as:

$$\ddot{x} = \frac{l\dot{\theta}^2\sin\theta - \dfrac{3}{4}g\sin\theta\cos\theta + \dfrac{u}{m}}{\dfrac{M}{m} + 1 - \dfrac{3}{4}\cos^2\theta} \tag{A.156}$$

Similarly, with negligible f_c, the relation (A.153) can be written as:

$$\ddot{\theta} = \frac{3}{4}\frac{\left[-\frac{u}{m}\cos\theta + \left(\frac{M}{m}+1\right)g\sin\theta - l\dot{\theta}^2\sin\theta\cos\theta\right]}{l\left(\frac{M}{m}+1-\frac{3}{4}\cos^2\theta\right)} \tag{A.157}$$

When θ is very small such that sin θ = θ, cos θ = 1 and $\theta\dot{\theta} = 0$, a linearised versions of Equations (A.156) and (A.157) can, respectively, be written as :

$$\ddot{x} = \frac{-\frac{3}{4}g\theta + \frac{u}{m}}{\frac{M}{m}+1-\frac{3}{4}} = \frac{-\frac{3}{4}g\theta + \frac{u}{m}}{\frac{M}{m}+\frac{1}{4}} \tag{A.158}$$

and

$$\ddot{\theta} = \frac{\left[-\frac{u}{m} + \left(\frac{M}{m}+1\right)g\theta\right]}{\frac{4}{3}l\left(\frac{M}{m}+\frac{1}{4}\right)} \tag{A.159}$$

A.15.1 The State Variable Representation

In order to obtain the state variable representation, we define the states as follows

$$\begin{aligned} x_1 &= \theta \\ x_2 &= \dot{\theta} \\ x_3 &= x \\ x_4 &= \dot{x} \end{aligned} \tag{A.160}$$

It is to be noted that the angle θ indicates the rotation of the pendulum rod about point P, and x is the location of the cart. The variables θ and x are taken as the outputs of the system

$$y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \theta \\ x \end{bmatrix} = \begin{bmatrix} x_1 \\ x_3 \end{bmatrix} \tag{A.161}$$

With the above choice of states and outputs, we can write the state variable representation of the nonlinear system, with negligible frictional force, by using the relations (A.156) and (A.157) whereas for linearised model the relations (A.158) and (A.159) will be used. We present below the linearised representation:

From the definition of the state variables and Equations (A.158) and (A.159). we can write the dynamics.

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= a_1 x_1 - bu \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -a_2 x_1 + \frac{41}{3}bu \end{aligned} \tag{A.162}$$

where $a_1 = \frac{(M+m)g}{md}$, $a_2 = \frac{gl}{d}$, $d = \frac{4l}{3}\left(\frac{M}{m}+\frac{1}{4}\right)$, and $b = \frac{1}{md}$

Therefore the state variable representation is given by:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ a_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ -a_2 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ -b \\ 0 \\ \frac{4l}{3}b \end{bmatrix} u \tag{A.163}$$

and
$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \quad \text{(A.164)}$$

Equations (A.163) and (A.164) are one form of state variable representations. Many other forms of state variable representation of the system may be obtained by making a different choice of the states.

Example A.2: Find the transfer function between the angular position and the control input of the linearised model of the inverted pendulum with M = 10 Kg, m = 1 Kg, l = 1m, g = 9.81 m/sec^2.

Solution : Substituting the values of the parameters given above in the equations (A.158) and (A.159), we have:

$$\ddot{x} = -0.7178\,\theta + 0.0976\,u \quad \text{(A.165)}$$

and
$$\ddot{\theta} = 7.8959\,\theta - 0.0732\,u \quad \text{(A.166)}$$

The transfer function between θ and the control input of the linearised system can be found from (A.166) as :

$$\frac{\theta(s)}{-\,U(s)} = \frac{0.0732}{s^2 - 7.8959} \quad \text{(A.167)}$$

The transfer function in Equation (A.167) clearly indicates an unstable system.

A.16 STEPPER MOTORS

Stepper motors are widely used in a variety of applications, the most notable of which is in the field of computer peripherals like printers, tape drivers, capstan drivers and memory access systems like hard drives. Stepper motors have also have been used in instrumentation and control associated with process control, machine tools and automatic analytic machines in medical services. In a *stepper motor* the rotor moves in discrete steps from one equilibrium position to the next. There are several general characteristics of a stepper motor that have made it the actuator of choice in such a large number of applications:

1. The device can be operated in *open-loop* configuration with a position accuracy of ±1 step within normal operating speed. Thus if a certain angular distance is specified, the motor can be commanded to rotate through an appropriate number of steps, and the mechanical elements coupled to the shaft will move through the required distance.
2. The motor exhibits high torque at small angular velocities. This is useful in accelerating a payload up to the rated speed starting from standstill position.
3. The motor exhibits a large holding torque with a dc excitation. Thus it has the property of being a "self-locking" device when the rotor is stationary. In fact, the rotor can move only when the terminal voltage *changes* with time.

In addition to these characteristics, there are other advantages that often make instrumentation and control engineers prefer the stepper motor over the dc servomotor. Some of these are :

(*a*) The stepper motor is directly compatible with digital controllers and/or computers without the use of digital to analog (D/A) converters.

(*b*) It provides excellent positioning accuracy, and errors when present are *non-cumulative*.

(*c*) Since open-loop control can be employed with the motor, it is often unnecessary to use a tachometer and/or an encoder resulting in considerable reduction of cost.

(*d*) The motor has a long and maintenance-free life and is thus a *cost-effective* actuator.

(*e*) The motor is rugged and it can be stalled without causing damage due to overheating.

The holding characteristics and relatively low cost compared with the dc servomotor makes the stepper motor a very attractive choice as an actuator for a robot joint. A very common fault like the shorted collector-emitter junction in a driver amplifier will make the robot-joint driven by a dc motor "run away" until an obstacle or mechanical limit-stop is encountered. A steeper motor driving a arm-joint, on the other hand, will move only *one step* and increase the *holding* force in presence of such a fault. So, the stepper motor, may help avoid an extremely dangerous situation and is found to be safer under this type of failure.

The open-loop-control feature, which contributes to its cost advantage over the DC servomotor, may sometimes be a problem. For example, if the manipulator encounters an unforeseen obstacle (like another machine tool in its workspace) during a move, the controller will nevertheless continue to output the calculated number of pulses to cause a specific arm motion. There is no feedback to know whether the arm has moved to the commanded position or not. A large position error will result if the arm is actually stalled due to obstruction. In such situations, position and/or velocity detector would be required, thereby neutralizing the cost advantage.

Large position errors can also be caused when attempts are made to move the rotor too fast. If the controller sends pulse sequence to the stators too rapidly, each pulse may not produce an actual step in the rotor shaft position. This is because of the fact that it takes a finite time for the current to grow in an inductive circuit like the stator and the associated flux to develop the motor torque for movement. This phenomenon of *dropped steps* will cause it to reduce the maximum rotor velocity.

Another potential difficulty with a stepper motor in a robot application is that the stepwise motion can excite significant manipulator oscillation because of the transient nature of the response. In absence of velocity feedback, oscillatory nature of response can be avoided by employing elaborate controller known as *microstepping* (vide section A.16.3). This further increases the cost of the drive electronics, eroding the cost differential between the stepper motor and servomotor.

A.16.1 Principles of Stepper Motor Operation

Two basic types of stepper motors can be constructed : (*a*) Variable reluctance (VR) type and (*b*) the permanent magnet (PM) type. Of the two varieties, currently the PM type has a wider range of applications. However, the functioning of the VR stepper motor will be considered in this section because the ease of its understanding.

Fig. A.47 shows the structure of a typical variable reluctance stepper motor. Both the stator and the rotor have *tooths* in contrast to that found in servomotors. The number of teeth in the rotor and stator must be different for the functioning. There are 8 equally spaced tooth in the stator (45° separation) and 6 equally spaced tooth in the rotor (60° separation) in the present illustration. Besides, each stator tooth has a coil wound on it and coils separated by 180° in space are grouped together as a single *phase*. For example, stator coils A and A′ forms a single phase and will be referred to as phase A. So, we have *phases* B (with stator coils B and B′), C (stator coils C and C′), and D(stator coils D and D′) respectively in the stepper motor shown in the figure.

(a) Rotor position when phase B is energised

(b) Rotor position when phase C is energised

(c) Rotor position when phase D is energised

(d) Rotor position when phase A is energised

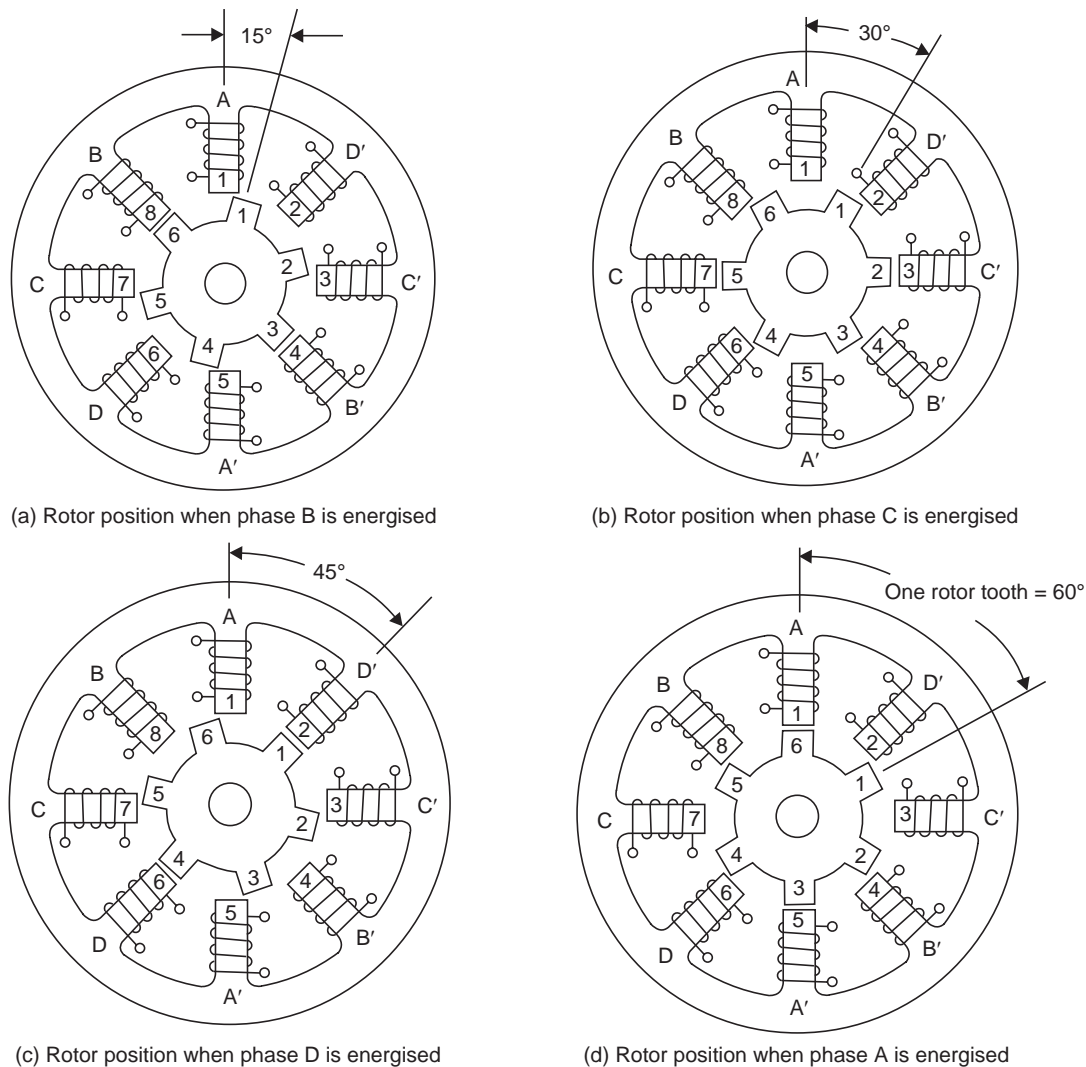



Fig. A.47 The variable reluctance stepper motor when excited in the sequence BCDA starting from the initial position where stator 1 is aligned with rotor 1 (consequently stator 5 is also aligned with rotor 4)

The operation of the variable reluctance stepper is based on the principle of "minimum reluctance" whereby a component forming a magnetic path always attempts to reorient itself so as to minimize the length of any air gap in the magnetic path. The magnetic device moves in such a way that the magnetic field can always find the path of *least resistance*. So, when phase B in Fig. A-47(*a*) is energized the rotor teeth 3 and 6 (R3, R6) will align with stator teeth 4 and 8 (S4, S8) and will remain in this position as long as the same phase remained excited. Under this condition, the motor has moved *one step* and is in a stable equilibrium condition as long as the excitation remains on coils B-B′. The excitation in the coil provides the *holding torque* and no motion will take place until a change in excitation occurs. Also, increasing the current through this phase will *not* cause the rotor to move but will increase this holding torque instead. Thus the locking force on the rotor will increase with increased excitation current. This should be contrasted with a servomotor, where increasing the excitation will increase the rotor speed.

Now when the excitation of phase B is stopped and phase C is excited, the rotor teeth R2, R5 will align with stator teeth S3, S7 as shown in Fig. A.47(*b*). So, the rotor is found to move clockwise through an angle of 15 degree (60 – 45 = 15). This process can be repeated for phases D, A, and then back to B (see figure). In each case, the rotor moves through a 15 degree step and with a complete sequence of phase excitations B, C, D, A and B will producing a rotation of 60 degrees. Therefore, it requires six such cycles to cause one complete rotor revolution of 360° and the stepper motor will be referred to as a "24-step/revolution" motor. It is to be noted that the sequence of excitation A, D, C, B and A will also produce 60 degree rotation. However, it can be easily verified that excitation sequence A, D, C, B, A will produce counterclockwise rotation of 60°. This sequence reversal can be easily implemented by programming on a microcomputer. In general, it can be demonstrated that at least three phases are required to make the rotation reversible.

The step angle can be expressed in terms of the number of stator teeth N_s and rotor teeth N_r as shown below:

$$\text{step angle} = 360° \frac{|N_s - N_r|}{N_s N_r} \tag{A.168}$$

and the number of steps/revolution is given by :

$$\text{number steps/revolution} = \frac{N_s N_r}{|N_s - N_r|}$$

Rotors of all stepper motors exhibit an underdamped transient response in moving from one step to another. As a result, an overshoot and undershoot about the equilibrium position may occur in a stepper motor. In certain applications, such behavior may not be acceptable (e.g., a robot), whereas in others it will be perfectly all right (e.g., a printer).

Stepper motors can be made with a wide range of steps/revolution. For economic reasons the upper limit is 200 steps which produces a step angle of 1.8°. For many applications, this relatively small angle is quite adequate. Other techniques can be used when finer step-angle resolution is required.

A.16.2 Half-Mode Operation

The method of energizing one phase at a time produces a step angle given by Equation (A.168) and is called *full-step mode*. Now suppose that two adjacent phases, say A and B, are energized *simultaneously*. An equilibrium point will be attained which is somewhere between the two full-step points as obtained by separately exciting phases A and B. As a matter of fact, with identical electrical properties of the coils in the two phases, if the same excitation amplitude is applied to both sets of coils, the new equilibrium point will be exactly halfway between the full-step points and the operation is known as *half-step mode*. This process can be repeated for phases BC, CD, and DA so that additional "halfway" equilibria can also be obtained. So, it is obvious that if the phase excitation sequence is A, AD, D, DC, C, CB, B, BA, A, the rotor will make twice the number of clockwise moves with respect to the full-step mode. Thus the stepper motor in Fig. A-47 will now have 48 discrete equilibrium points per revolution. Since the rotation angle per step has been approximately cut to half, the angular overshoot of the rotor in moving from point to point is also reduced. Reversing the phase excitation sequence will cause the rotor to rotate in the counterclockwise direction. The switching circuitry needed to produce half-step operation is somewhat more costly than the relatively simple full-step electronics.

A.16.3 Microstep Mode

It is possible to use an excitation voltage anywhere between 0 and the rated value V in order to energize the second phase of the stator coil. This scheme of operation generally referred

to as *micro-stepping* mode. In order to decrease computational burden on the stepper motor controller, the micro-step size is generally determined by dividing the angular distance of a full step by an integral power of 2 (e.g., 2, 4, 8, 16, or 32). Implementation of micro-stepping requires considerably more complex switching circuitry, so it will be costlier compared to that needed for a full-step operation. However, in a robot application it is preferred since oscillation at the desired final point is usually smaller.

A.17 PERMANENT-MAGNET STEPPER MOTORS

Permanent Magnet motor consists of a multi-phased stator and a two-part, permanent-magnet rotor both of them are also toothed (see Fig. A-48), as with the variable reluctance stepper motor. However, the opposite ends of the rotor in the present case are north and south poles of a permanent magnet with the teeth at these ends being offset by half a tooth pitch (Fig. A.48*b* and *c*). The permanent magnet stepper motor can also be operated in full, half, or microstep mode. The major differences between the two classes of stepper motors are depicted in Table A.3.

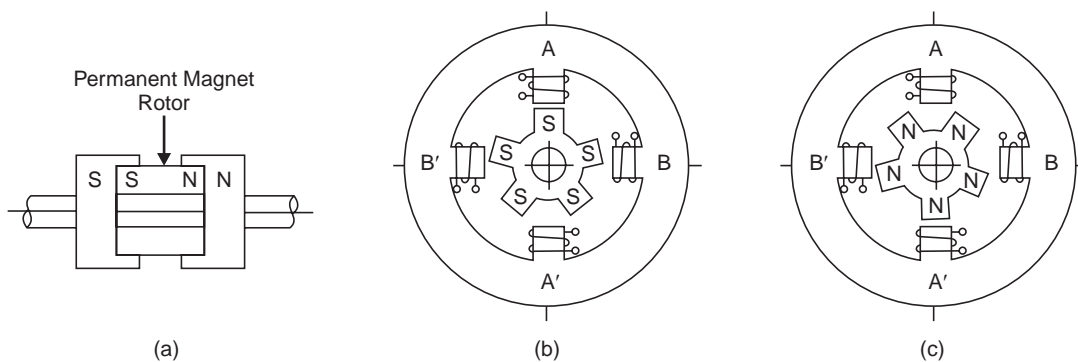



Fig. A.48 Permanent magnet stepper motor

The permanent magnet stepper rotor position is dependent on the polarity of the phase excitation so that, a *bipolar* signal is required to achieve bi-directional control. This can be done by using only a pair of phases as shown in Fig. A.48, whereas the equivalent variable reluctance stepper required four phases (see Fig. A.47).

Using a bipolar power supply, the motor in Fig. A-48 can be driven in the full-step mode with the switching scheme shown in Fig. A.49*a* employing two *tristate* switches, S_1 and S_2. In the figure, phase A is connected to positive supply while phase B is connected to off position. A possible method of realising such a switching circuit is shown in Fig. A-49*b*. The "fly-back" diodes (D_1-D_4) are normally used to protect the non-conducting power transistors from the "inductive kick" that occurs when the current through an inductor is suddenly interrupted. Without this protection, double the supply voltage will be impressed across the collector-emitter junction of a non-conducting transistor, which might be well in excess of its breakdown voltage during the switching interval.

For the motor in Fig. A.48, it can be shown that each step is 18° and that there will be 20 steps/revolution. The excitations and simple control signals to the transistors that produce four rotor steps (i.e., 72°) are shown in Fig. A.50.

In actual operation, a microprocessor would determine the number of steps needed to cause a load to be moved a certain distance. This would be done for each joint in a robot

application. The processor would then transmit the information, together with direction and step timing data, to a discrete digital hardware package. The latter would keep track of the total number of steps moved and would implement the appropriate switching sequence. Clearly, this would represent open-loop joint control, with its inherent problems. Closed-loop control can also be implemented by using position feedback. However, this would negate some of the cost advantage of the stepper motors over the servomotor.

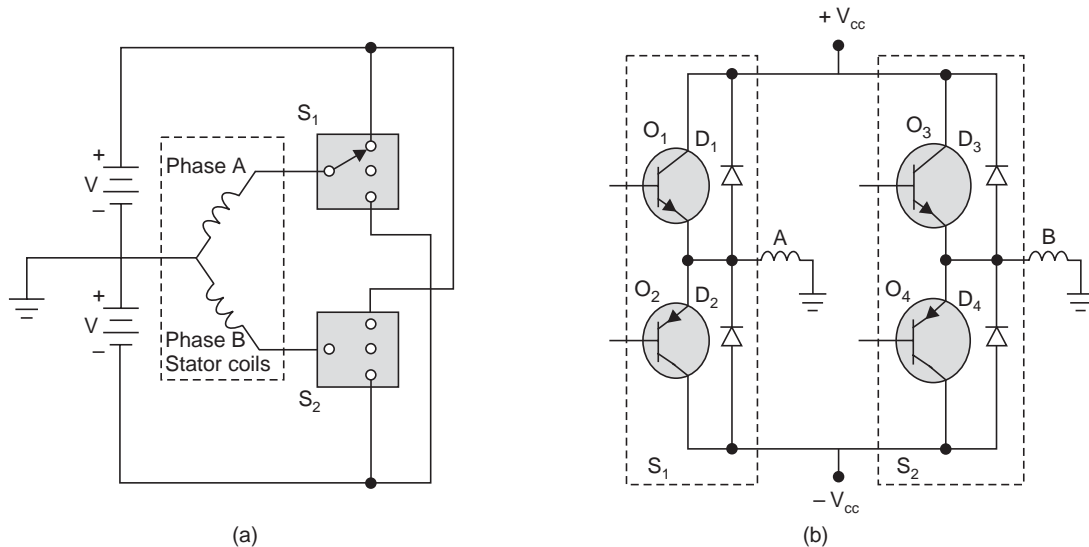



Fig. A.49 Switching circuit for bi-directional rotation of permanent magnet stepper motor

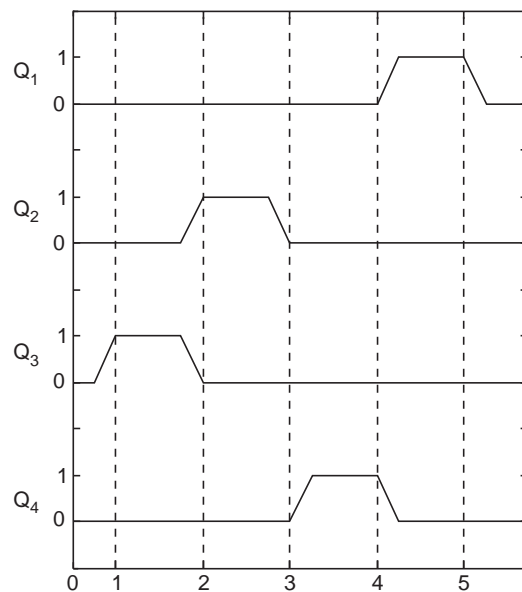



Fig. A.50 Pulse sequence applied to transistor switches in Fig. A-48(*b*) for rotation of 4 steps of stepper motor

Table A.3 Differences between permanent magnet and variable reluctance stepper motors

Characteristics type	*Permanent magnet motor*	*Variable reluctance motor*
1. Motor	It is magnetized	It is not magnetized
2. Rotor position	Depends on stator excitation polarity	Independent of stator excitation
3. Rotor inertia	High due to magnet	Low (in absence of a magnet)
4. Mechanical response	Not so good (due to high inertia)	Good (low-inertia device)
5. Inductance	Low due to rotor offset	Generally high for same torque
6. Electrical response	Faster current rise (due to low inductance)	Slower current rise (due to higher inductance)

A.18 CONTROL OF A LOAD THROUGH A FLEXIBLE LINK

Many mechanical systems such as a robot arm or a disk drive read-write head assembly would exhibit some degree of structural flexibility, which can be modeled schematically by double mass-spring device as shown in Fig. A.51. Even the antenna assembly tracking a satellite may have some flexibility between the angle observed by the antenna and the angle of the base. We shall obtain representation of the dynamics of such a system, both in the state variable form and transfer function form.

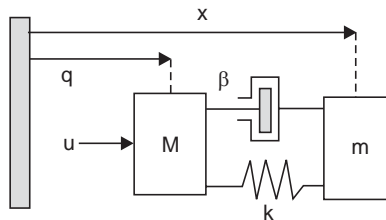



Fig. A.51 Two bodies of mass M and *m* respectively coupled through a damper and spring

With reference to Fig. A.51, we use the following terminologies :

M, m = masses of coupled bodies

β = viscous friction coefficients

k = spring constant

u = applied force

x and q are outputs

The equations of motion may be written as:

$$M\ddot{q} + (\dot{q} - \dot{x})\beta + (q - x)k = u$$
$$m\ddot{x} + (\dot{x} - \dot{q})\beta + (x - q)k = 0 \quad \text{(A.169)}$$

With choice of states $x_1 = x,\ x_2 = \dot{x}_1 = \dot{x},\ x_3 = q,\ x_4 = \dot{x}_3 = \dot{q}$, the above equation can be written in the state matrix form:

$$\dot{X} = AX + bu$$

$$Y = cX + du$$

where X is a 4 × 1 state vector, $X = [x_1\ x_2\ x_3\ x_4]'$ and Y is a 2 × 1 outpt vector $Y = [y_1\ y_2]'$

The matrices are given by

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k/m & -\beta/m & k/m & \beta/m \\ 0 & 0 & 0 & 1 \\ k/M & \beta/M & -k/M & -\beta/M \end{bmatrix},\quad b = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix},\quad c = \begin{bmatrix} 1 & 0 & 1 & 0 \end{bmatrix} \text{ and } d = 0.$$

Taking the Laplace transforms of Equation (A.169) with zero initial conditions, we can find out the transfer functions between u and the two outputs and the result is given below :

$$\frac{Y_1(s)}{U(s)} = G_2(s) = \frac{1}{M}\frac{\frac{\beta}{m}s + \frac{k}{m}}{s^2\left[s^2 + \left(1 + \frac{m}{M}\right)\left(\frac{\beta}{m}s + \frac{k}{m}\right)\right]} \tag{A.170}$$

and

$$\frac{Y_2(s)}{U(s)} = G_1(s) = \frac{1}{M}\frac{s^2 + \frac{\beta}{m}s + \frac{k}{m}}{s^2\left[s^2 + \left(1 + \frac{m}{M}\right)\left(\frac{\beta}{m}s + \frac{k}{m}\right)\right]} \tag{A.171}$$

The second transfer function represents the case where the position sensor is placed on the same rigid body as the actuator, but the dynamics of system is affected by mechanical oscillation from other structures in the system that is coupled to the mass to which the actuator and sensor are attached.

The first transfer functions represents the case where there is structural flexibility between the position sensor and the actuator.

Mapping from S to Z Domain

B.1 MAPPING OF S DOMAIN POLES TO Z DOMAIN

If the Laplace transform of a continuous function $f(t)$ be represented by F(s) before sampling and by F*(s) after it is sampled, then F*(s) can be written, from relation (10.2) of Chapter 10, as :

$$F^*(s) = £\,[f^*(t)] = f(0) + f(\mathrm{T})e^{-\mathrm{T}s} + f[(2\mathrm{T})]e^{-2\mathrm{T}s} + \ldots\ldots$$

$$= \sum_{k=0}^{\infty} f(k\mathrm{T})e^{-k\mathrm{T}s} \tag{B.1}$$

where T is the sampling period and $f(k\mathrm{T})$ is the value of the continuous function at time $t = k\mathrm{T}$. The angular sampling frequency ω_s is given by $2\pi/\mathrm{T}$.

Now if F(s) has a pole located at $s = s_1$, then F*(s) will be found to have poles located at $s = s_1 + jn\omega_s$ where $n = 0, \pm1, \pm2, \ldots \pm \infty$. This can be shown by replacing s with $s_1 + jn\omega_s$ in relation (B.1)

$$F^*(s_1 + jn\omega_s) = \sum_{k=0}^{\infty} f(k\mathrm{T})e^{-k\mathrm{T}(s_1 + jn\omega_s)} \quad \text{where } n \text{ is any integer including 0.}$$

Since $\qquad \omega_s = \dfrac{2\pi}{\mathrm{T}}$, we have $e^{-jkn\omega_s \mathrm{T}} = e^{-j2\pi kn} = 1$

Therefore,

$$F^*(s_1 + jn\omega_s) = \sum_{k=0}^{\infty} f(k\mathrm{T})e^{k\mathrm{T}s_1}e^{-jkn\omega_{s1}\mathrm{T}} = \sum_{k=0}^{\infty} f(k\mathrm{T})e^{-k\mathrm{T}s_1} = F^*(s_1) \tag{B.2}$$

Equation (B.2) implies that, a pole (zero) of F(s) at $s = s_1$ in the s plane is mapped to multiple poles (zeros) of the sampled function F*(s) located at $s = s_1 \pm jn\omega_s$. Fig. B.1 shows the periodic function F*(s) in the s-plane which is segregated in strips of width $\omega_s/2$, where F(s) has a pole at $s = s_1$ and ω_s is the angular sampling frequency. The strip bound by $\pm\,\omega_s/2$ is referred to as **Primary Strip** (vide Fig. B.1), whereas others are referred to as **Complimentary Strips**. In the analysis and synthesis of a control systems only the poles in the primary strips are considered, ignoring the poles in the complementary strips whose contributions in the frequency response are negligibly small because of the low-pass filtering characteristic of the continuous plant together with the ZOH device preceding the plant. However, it is very interesting to note that the primary pole s_1 (zero) along with all it's associated complementary poles (zeros) map onto the same point in the z-plane through the relation

$$z = e^{s\mathrm{T}} \tag{B.3}$$

B.1.1 Mapping of the Primary Strip

Let us now map the boundary *oabcde* of the left-half portion of primary strip in the s-plane of Fig. B.2(a) into the z-plane using the defining relation of (B.3).

$$z = e^{sT} = e^{(\sigma \pm j\omega)T} = e^{\sigma T}\, e^{\pm j\omega T}$$

$$= e^{\sigma T} \angle \pm \omega T \tag{B.4}$$

We consider the contribution of the various segments of the boundary *oabcde* as we traverse in the counter clockwise direction, as shown below :

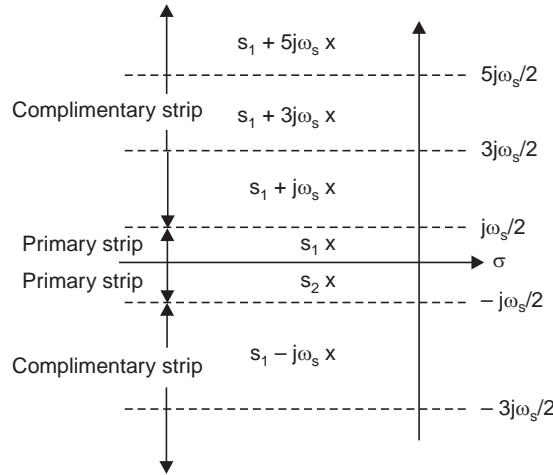



Fig. B.1 Location of the poles of F*(s) corresponding to the pole s_1 of F(s)

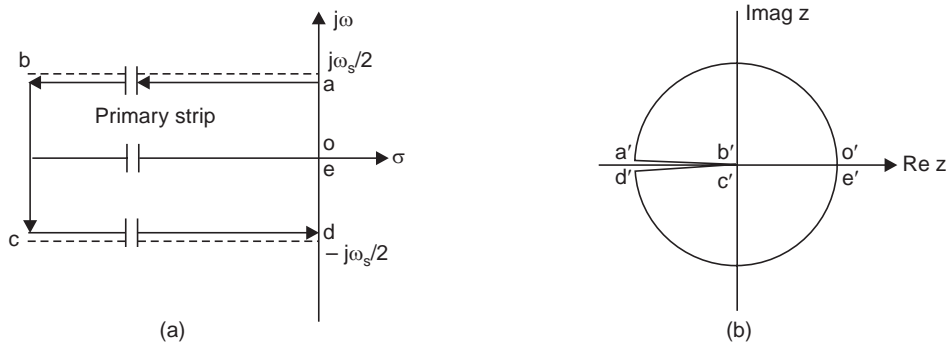



Fig. B.2 The primary strip in the left hand s-plane mapped into inside z-plane

1. Interval [o, a]

At point o, $\sigma = 0$, $\omega = 0$, so from (B.4) we have $0'$ at $z = 1$, $\angle 0°$

Similarly at point a, $\sigma = 0$, $\omega \to \omega_s/2$, therefore a is mapped to a' at $z = 1$, $\angle 180°^{-}$

Proceeding in this way, we evaluate the contributions at the extreme points of the following segments.

2. Interval [a, b]

at point b, $\sigma \to -\infty$, $\omega \to \omega_s/2$, $b' \to z = 0$, $\angle 180°^{-}$

3. Interval [b, c]

at point c, $\sigma \to -\infty$, and $\omega \to -\omega_s/2$, $c' \to z = 0, \angle -180^{\circ-}$

4. Interval [c, d]

at point d, $\sigma = 0$, and $\omega \to -\omega_s/2$, $d' \to \quad z = 1, \angle -180^{\circ-}$

5. Interval [d, e]

at point e, $\sigma = 0$, and $\omega = 0$, $e' \to z = 1, \angle 0°$

The expression $\angle 180^{\circ-}$ in the above relations, stands for an angle which is slightly less than 180°.

The points o a b c d and e on the boundary of the primary strip *oabcde* in the left half s-plane, therefore, maps to points o', a', b', c', d' and e' lying on a unity circle, with center at the origin in z-plane with a rounded slot on the negative real axis, as shown in Fig. B.2(b). The boundary a' b' c' d' in the limit, approach the portion of the negative real axis lying between 0 and the point $-1 + j0$ in z-plane. So, for convenience, the slotted figure in the limit, is usually referred to as a unit circle.

B.1.2 Mapping of the Constant Frequency Loci

The mapping of the constant frequency $\omega = \omega_1 < \omega_s/2$ in the s-plane into z-plane is expressed by $z = e^{\sigma T} \angle \omega_1 T$. So, any constant frequency loci in the s-plane, which is a horizontal line, is mapped into a radial line in the z-plane emanating from the origin with a angle of $\omega_I T$ rad with the real axis as shown in Fig. B.3.

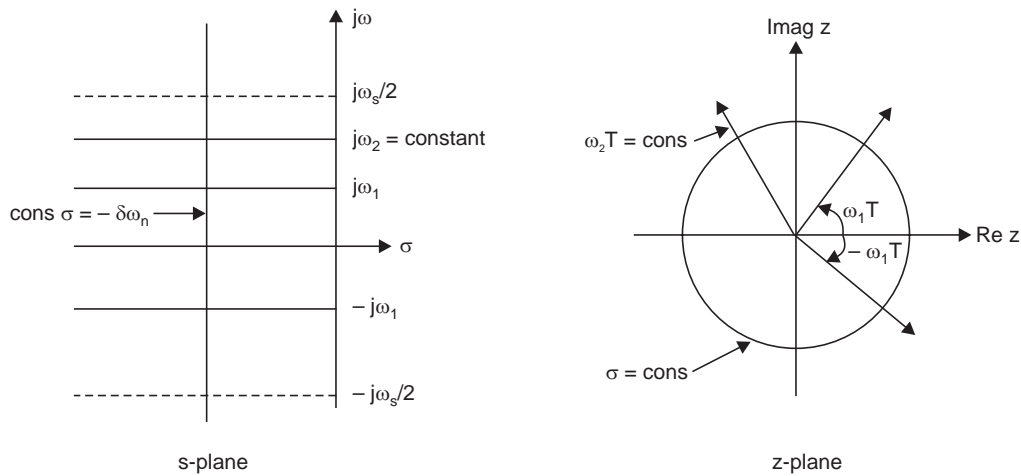



Fig. B.3 Mapping of the constant frequency loci from s-plane into the z-plane

B.1.3 Mapping of the Constant Damping Co-efficient Loci

The settling time for a second order system within 2% band of its steady state output when subjected to a step input is given by $t_s \approx 4/\delta\omega_n$. Therefore, when the roots lie on the vertical line in the s-plane, $\sigma = -\delta\omega_n$ = constant, the settling time will remain constant. The mapping of this constant s-plane damping-coefficient loci, $\sigma = \sigma_1$ of the second order quadratic pole into the z-plane is expressed by $z = e^{\sigma_1 T} \angle \omega_1 T$. We have noted in sec B.1.1 that the line segment of the s-plane on the imaginary axis from $-j\frac{\pi}{T}$ to $\frac{j\pi}{T}$ are mapped onto the circle

boundaries in the z-plane. Therefore, any constant damping coefficient locus parallel to imaginary axis is mapped onto a circle of radius $|z| = e^{\sigma_1 T}$ centered at the origin of the z-plane. If $\sigma_1 < 0$, then this circle will lie within the unit circle and if $\sigma_2 > 0$, the corresponding circle will lie outside the unit circle.

B.1.4 Mapping of Constant Damping Ratio Line and Constant Natural Frequency to z-Domain

We have observed in Chapter 4 [vide section 4.5.2.1 Equation (4.14)] that the percent over shoot of a prototype second order system subjected to a step input solely depends on the damping ratio. Also we know that the rise time is inversely proportional to bandwidth of the system. So, it will be of interest to us to map these parameters to the z-domain through the relation (B.4). The damping ratio of the dominant z-plane roots is very often expressed in terms of the *equivalent damping ratio* of the dominant roots in the s-plane.

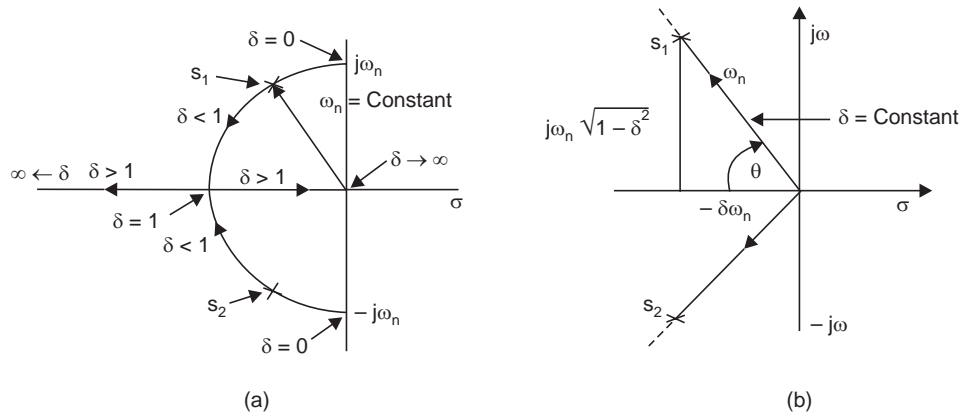



Fig. B.4 Constant damping ratio line and constant frequency

We know the roots of the characteristic equation of the proto type second order system are given by:

$$s_{1,2} = \sigma + j\omega_d = -\,\delta\omega_n \pm j\omega_n\sqrt{1-\delta^2} \tag{B.5}$$

The location of the root s_1 is shown in Fig. B.4. Denoting the angle the radial line from the origin to the pole s_1 makes with the negative real axis by θ, we have

$$\delta = \cos\theta \tag{B.6}$$

and
$$\tan\theta = \frac{\omega_n\sqrt{1-\delta^2}}{\delta\omega_n} = \frac{\omega_d}{\delta\omega_n} = \frac{\sqrt{1-\delta^2}}{\delta} \tag{B.7}$$

or,
$$\delta\omega_n = \frac{\omega_d}{\tan\theta} = \omega_d \cot\theta, \text{ where } \omega_d = \omega_n\sqrt{1-\delta^2}$$

Therefore,
$$s_1 = \sigma + j\omega_d = -\,\delta\omega_n + j\omega_d = -\,\omega_d\cot\theta + j\omega_d$$

and the s-domain pole mapped to the z-domain becomes

$$z = e^{s_1 T} = e^{(-\,\omega_d \cot\theta + j\omega_d)T} = e^{-\,2\pi\omega_d \cot\theta/\omega_s} \angle 2\pi\omega_d/\omega_s \tag{B.8}$$

The relation (B.6) shows that when δ is constant, θ is also constant. So the locus of constant damping ratio is a line emanating from the origin in the s-plane making an angle $\theta = \cos^{-1}(\delta)$ with the negative real axis as shown in Fig. B.4(b).

We note from Equation (B.7) that for a given constant value of $\delta = \delta_1$, cot θ is constant. So, z in Equation (B.8) is a function of ω_n (and hence ω_d) only for a given sampling frequency. Therefore, the constant damping-ratio locus in the s-plane maps into the z-plane as a logarithmic spiral (except for θ = 0 and 90°). The logarithmic spiral for a constant δ line undergoes one-half revolution in the z-plane as ω_d is varied from 0 to $\frac{\omega_s}{2}$. It undergoes another one-half revolution for $\omega_n = 0$ to $\omega_d = -\frac{\omega_s}{2}$. The plot of the logarithmic spiral with δ set to various constant values from 0 to 1 is shown in Fig. B.5 as ω_n is varied from 0 to π in steps of π/10T, with sampling period T = 1 sec.

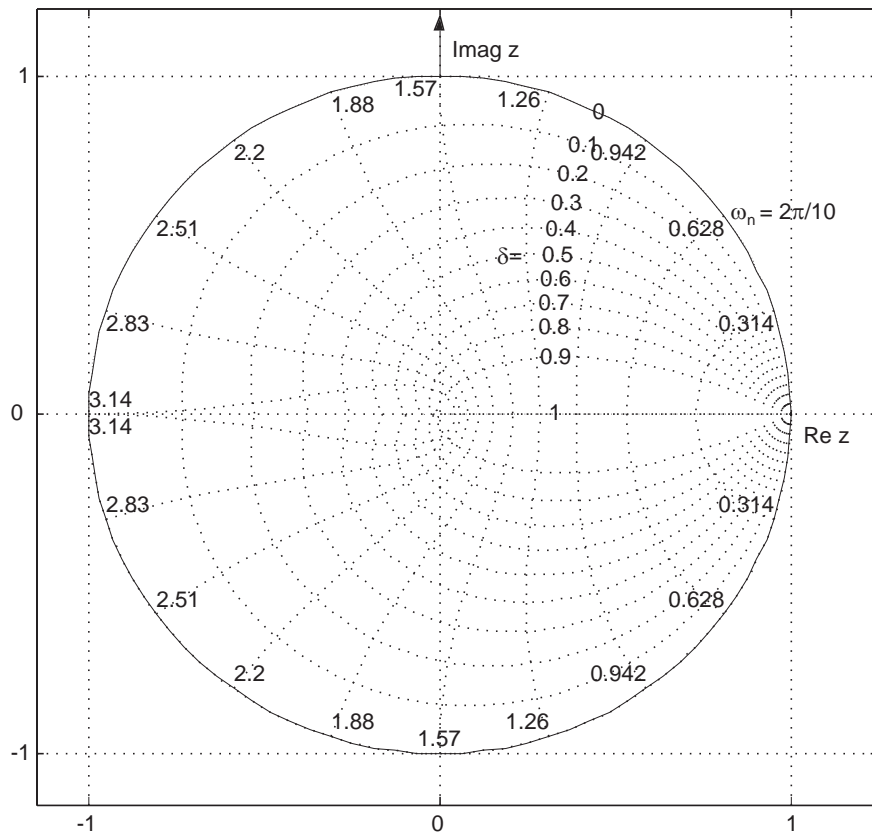



Fig. B.5 z-plane loci of constant damping ratio δ and undamped frequency ω_n with sampling perod T =1 sec.

Substituting the values of $\delta = 1$, $\omega_d = 0$ and $s_1 = -\mid \sigma \mid$, corresponding to the roots lying on the negative real axis in the s-plane, in relation (B.8) we get roots on the positive real axis in the z-plane that lie within the unity circle. Also we can verify that complex roots in the primary strip (excluding the boundary) of the left-half s-plane map onto complex roots inside the unity circle of the z-plane. The roots lying on the boundary line of the left-half primary strip (as well as complimentary strips) will map onto roots lying on the negative real axis inside unity circle in the z-plane (see Example B.1 below).

Further, it can be shown that each complementary strip in the left-half s-plane maps onto the unity circle in the z-plane while the strips in the right half s-plane, primary as well as complimentary, each map onto the entire z-plane excluding the area covered by the unity circle. It will be observed that the roots on positive real axis in the s-plane with $\delta = -1$, $\omega_d = 0$ and $s_1 = + \mid \sigma \mid$ will map onto roots lying on the non-negative real axis in the z-plane outside unity circle.

From Fig. B.5, it is observed that the straight line nature of constant damping ratio and semicircular nature of constant ω_n found in the primary strip of s-plane are preserved in the region close to $z = +1$. The constant δ and constant ω_n plots are superimposed on the root locus in the z-plane to locate roots of the closed loop system for a desired output response.

The MATLAB command *zgrid*(z, ω_n) will give the constant δ and constant ω_n plot for a given z and ω_n vector-pairs.

Example B.1 We illustrate the mapping of s-plane poles and zeros of F(s) given below to z-domain poles and zeros

$$F(s) = \frac{s + \sigma_1}{(s + \sigma_1)^2 + \omega_1^2}, \text{ with sampling frequency } \omega_s = 2\pi/T$$

We identify the above function as the Laplace transform of $e^{-at} \cos \omega t$ (entry number 21 in Table 2.1 of chapter 2) whose z-transform can be written from table 10.1 (entry number 24 in Chapter 10) as :

$$F(z) = \frac{z^2 - ze^{-\sigma_1 T} \cos \omega_1 T}{z^2 - 2ze^{-\sigma_1 T} \cos \omega_1 T + e^{-2\sigma_1 T}}$$

Now if we choose $\omega_1 = \omega_s/2$, so that $\cos \omega_1 T = \cos \pi = -1$, then F($z$) becomes

$$F(z) = \frac{z(z + e^{-\sigma_1 T})}{(z + e^{-\sigma_1 T})^2} = \frac{z}{(z + e^{-\sigma_1 T})}$$

The poles and zeros of F(s) and F(z) are shown in Fig. B.6. We notice that two poles and one zero of F(s) is mapped to one pole and one zero in F(z). We also notice that the pole lying on the boundary of the primary strip in the left-half s-plane is mapped to the negative real axis in z-plane inside the unity circle.

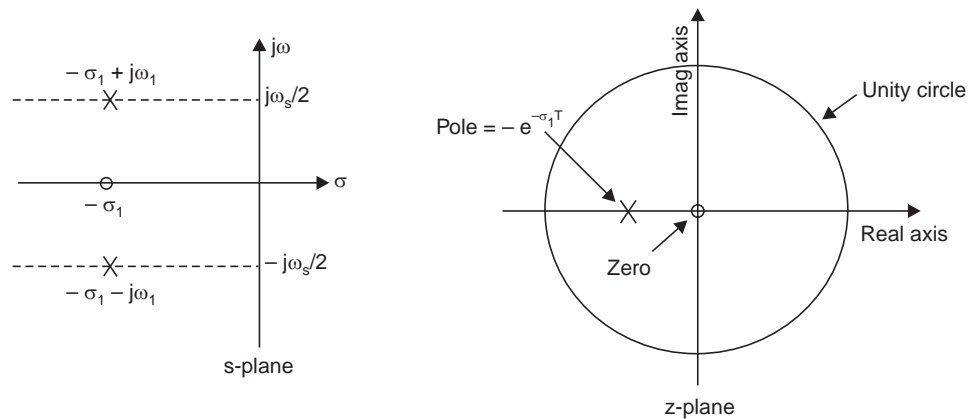



Fig. B.6 Mapping of *s*-plane pole-zeros to *z*-plane poles-zeros

Discretization of Analog Controllers

C.1 BILINEAR TRANSFORMATION

In Section 10.14 of Chapter 10 we considered a bilinear transformation $\lambda = \dfrac{z-1}{z+1}$ (C.1)

which allowed us to use Routh-Hurwitz criteria to investigate stability in the λ domain. By slightly modifying the transformation in relation (C.1) we can adapt the bilinear transformation for designing compensators in a continuous domain and transfer the result to z-domain.

Let us consider the bilinear Tustin transformation $w = \dfrac{2}{T}.\dfrac{z-1}{z+1}$ (C.2)

such that
$$z = \frac{2 + Tw}{2 - Tw} \tag{C.3}$$

Since $z = e^{sT}$, the relation (C.2) may be used to establish a relationship between s and w-plane as shown below:

$$w = \frac{2}{T}.\frac{z-1}{z+1} = \frac{2}{T}.\frac{e^{sT}-1}{e^{sT}+1} = \frac{2}{T}.\frac{e^{sT/2} - e^{-sT/2}}{e^{sT/2} + e^{-sT/2}}$$

or
$$w = \frac{2}{T}.\tanh(sT/2) \tag{C.4}$$

Now for points on the imaginary axis in the s-plane, we have $s = j\omega$. Substituting this value of s in Equation (C.4), we get

$$w = j\frac{2}{T}.\tan(\omega T/2) = j\omega_w \text{ (say)} \tag{C.5}$$

where ω_w is the imaginary part of w.

Now for points on the real axis in the s-plane, $s = \sigma$. Putting this value of s in relation (C.4), we get

$$w = \frac{2}{T}.\tanh(\sigma T/2) = \sigma_w \text{ (say)} \tag{C.6}$$

The relations (C.5) and (C.6) map z-plane to w plane with a one to one correspondence of points (conformal mapping).

The mapping of the s-plane poles and zeros into w-plane by use of relations (C.5) and (C.6) is referred to as *prewarping of s-plane poles and zeros.*

The bilinear transformation of Equation (C.2) is found to perform the following mapping

(*i*) The interior of unity circle to the left-half plane of w

(*ii*) The outside of unity circle is mapped to the right-half plane of w-plane

(*iii*) The unity circle on the imaginary axis of w-plane

If the sampling period is such that $\omega T/2 < 0.3$ and $\sigma T/2 << 1.41$ then $\tan(\omega T/2) = \omega T/2$ and $\tanh(\sigma T/2) = \sigma T/2$, respectively, in which case we get from Equations (C.5) and (C.6)

$$\omega = \omega_w \quad \text{and} \quad \sigma = \sigma_w \tag{C.7}$$

So, when the approximation is valid, we can write, $w = \sigma_w + j\omega_w = \sigma + j\,\omega = s$ (C.8)

We also note from relation (C.5) that the imaginary axis in the primary strip in the s-plane is mapped to the entire imaginary axis in the w-plane. The w-plane is found to be very similar to the s-plane and the analysis and design methods can be applied to the w-plane *as if the w-plane model represents continuous system*.

Thus, the relation (C.8) forms a basis of design concept, where compensator is designed in the s-domain and the result is transferred to the w-domain and, by means of relation (C.2), to z-domain for realization of the compensator on a microcomputer.

C.2 FREQUENCY DISTORTION IN BILINEAR TRANSFORMATION

The bilinear transformation does not preserve the frequency response. This is apparent from the following discussions.

Consider a compensator in the w-domain : $D(w) = u(w)/e(w)$

Using the bilinear transformation we get the compensator in the z-domain as

$$D(z) = D(w)\big|_{w = (2/T)(z-1)(z+1)} \tag{C.9}$$

We shall compare the responses $D(j\omega_w)$ to $D(e^{j\omega})$ where $D(\omega_w)$ is the frequency response of the continuous system $D(w)$, and $D(e^{j\omega})$ is the frequency response of the discrete system $D(z)$. In order to get the frequency response, s is replaced by $j\omega$ and z is replaced by $e^{j\omega}$ in the bilinear transformation to get the following result [see Equation (C.5)]

$$\omega_w = \frac{2}{T}.\tan(\omega T/2) \tag{C.10}$$

Equation (C.10) relates the distortion between ω_w and ω, which is plotted in Fig. C.1

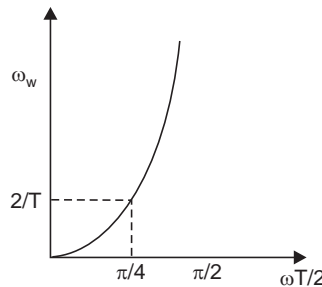



Fig C.1 Relationship between ω_w and ω

From relation (C.9) and (C.10) we obtain :

$$|D(w)|_{w = j\omega_w} = |D(z)|_{z = e^{j\omega T}} \tag{C.11}$$

$$\text{if} \qquad \omega_w = (2/T)\tan(\omega T/2)$$

Since the amplitude of $D(w)$ at frequency ω_w is equal to the amplitude of $D(z)$ at frequency ω where $\omega_w = (2/T)\tan(\omega T/2)$, the frequency responses in the w domain is distorted when mapped in the z-domain. This distortion is depicted in Fig. C.2. One strategy to take care of the

distortion present in the frequency response of the compensator is to account for it in the design stage itself.

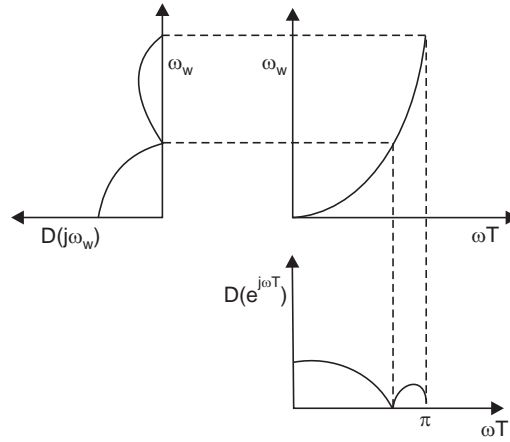



Fig. C.2 Distortion of frequency response caused by the bilinear transformation

It is evident from Fig. C.2 that the bilinear transformation compresses the continuous frequency $0 < \omega_w < \infty$ to a limited digital range $0 < \omega T < \pi$. In the design stage of the compensator, the continuous poles and zeros are prewarped before hand. The bilinear transformation will map them back to the designed poles and zeros in the discrete domain. The adjustment can be done in a two step procedure as is illustrated below:

Step 1. Replace all poles and zeros of D(s) by their prewarped values. Poles and zeros appear as factors $(s + \alpha)^{\pm 1}$ in D(s). For instance, a factor $(s + \alpha)$ in D(s) is replaced as shown below:

$$s + \alpha = (s + \alpha_1) \mid \alpha_1 = (2/T) \tan (\alpha T/2) \tag{C.12}$$

Step 2. Transform D(s, α_1) to D(z, α_1)

$$D(z, \alpha_1) = D(s, \alpha_1) \mid s = (2/T)(z - 1)/(z + 1) \tag{C.13}$$

It is to be noted the dc condition of the compensator is to be satisfied separately.

Example C.1 Let us consider the example:

$$D(s) = \alpha/(s + \alpha)$$

Step 1. Prewarp the poles and zeros. For this simple example, there is only one pole located at $s = -\alpha$

So, $D(s, \alpha_1) = \alpha/[s + \{(2/T) \tan (\alpha T/2)\}]$

Step 2. Substituting $s = 2(z - 1)/T(z + 1)$ yields

$$D(z) = \frac{k\alpha}{\frac{2}{T}\frac{z-1}{z+1} + \frac{2}{T}\tan\frac{\alpha T}{2}} \tag{C.14}$$

Now the dc value of $D(s) = \lim_{s\to 0} D(s) = D(0) = 1$.

Now setting the dc value of D(z) to 1, we get,

$$\lim_{z\to 1} D(z) = \lim_{z\to 1} \frac{k\alpha}{\frac{2}{T}\frac{z-1}{z+1} + \frac{2}{T}\tan\frac{\alpha T}{2}} = \frac{k\alpha}{\frac{2}{T}\tan\frac{\alpha T}{2}} = 1$$

Therefore,
$$k = \frac{\frac{2}{T}\tan\frac{\alpha T}{2}}{\alpha} \tag{C.15}$$

From Equations (C.14) and (C.15), we have

$$D(z) = \frac{\tan\frac{\alpha T}{2}}{\frac{z-1}{z+1} + \tan\frac{\alpha T}{2}} = \frac{z+1}{z + z\cot\frac{\alpha T}{2} + 1 - \cot\frac{\alpha T}{2}}$$

Some important properties of the bilinear transformation and frequency prewarping are listed below:

(*i*) It maps the L.H.S of the *s*-plane to unit circle on the *z*-plane (one to one correspondence).

(*ii*) A stable D(*s*) transforms to a stable D(*z*).

(*iii*) There is no aliasing.

(*iv*) It matches the frequency response for breakpoints and for zero-frequency and compresses the response at $\omega_w = \infty$ to $\omega T = \pi$.

(*v*) The impulse response and phase responses are not preserved.

Notes on MATLAB Use

MATLAB is a high performance language for technical computing [57]. It integrates computation, visualization and programming in an easy to use environment where problems and solutions are expressed in familiar mathematical notation. MATLAB is an interactive system whose basic data element is an array that does not require dimensioning.

The MATLAB stands for *matrix laboratory*. It features a family of application specific solutions called *toolboxes.*

The user communicates with the MATLAB through Command Window. The MATLAB Command Window displays the prompt **>>** indicating that it is waiting for the command from the user.

In order to create a simple vector with 9 elements with assigned name 'x' enter the following line after the prompt >> and then press the enter key

>> x = [1 2 3 4 5 4 3 2 1]   %x is a 9 × 1 row vector.

The symbol % is the comment in MATLAB. The percentage symbol % is not interpreted for execution. The MATLAB responds as :

```
x =
  1   2   3   4   5   4   3   2   1
```

In order to add 1 to each element of the vector, '*x*', and store the result in a new vector '*y*', type in :

```
>> y = x + 1
y =
  2   3   4   5   6   5   4   3   2
```

In order to plot the result of the vector addition with grid lines type in :

```
plot(y)
grid on
```

MATLAB can use symbols in plots. Here is an example using '+'s to mark the points.

MATLAB offers a variety of other symbols and line types.

```
plot(y,'+')
axis([0 10 0 8])
```

One area in which MATLAB excels, is matrix computation. Creating a matrix is as easy as making a vector, using semicolons (;) to separate the rows of a matrix.

```
A = [1 2 0; 2 5 –1; 4 10 – 1]
A =
   1    2    0
   2    5   – 1
   4   10   – 1
```

We can easily find the transpose of the matrix A.

```
B = A′
B =
    1    2    4
    2    5   10
    0   –1   –1
```

These two matrices can be multiplied as :

```
C = A * B
C =
    5   12   24
   12   30   59
   24   59  117
```

It is to be noted that MATLAB doesn't require the user to deal with matrices as a collection of numbers. Instead of doing a matrix multiplication, we can multiply the corresponding elements of two matrices or vectors using the.* operator.

```
C = A.* B
C =
    1    4    0
    4   25  –10
    0  –10    1
```

The inverse of a matrix is found as

```
X = inv(A)
  X =
    5    2   –2
   –2   –1    1
    0   –2    1
```

A matrix when multiplied by its inverse, the result is identity matrix.

```
I =X * A
I =
    1    0    0
    0    1    0
    0    0    1
```

The "poly" function generates a vector containing the coefficients of the characteristic polynomial.

The characteristic polynomial of a matrix F is det(lambda*I - F)

```
>> F = [0 1; –5 –6]
p = round(poly(F))
p =
    1    6    5
```

We can easily find the roots of a polynomial using the "roots" function. These are actually the eigenvalues of the original matrix.

>> roots(p)

ans =

– 5

– 1

Transfer function g with zero located at – 2 and two poles located at – 1 and – 5 and forward path gain K = 1 is found from command zpk

>> g = zpk([– 2], [– 1 – 5], 1)

Zero/pole/gain:

$$\frac{(s + 2)}{(s + 1)(s + 5)}$$

Suppose we are interested to express the denominator of transfer function g in the polynomial form. First write the coefficients of the factor s + 1 in descending power of s as $d1$

>> $d1$ = [1 1];

similarly we write the other factor as $d2$.

>> $d2$ = [1 5];

Now performing convolution on $d1$ and $d2$ we get the denominator 'den'

>> den = conv($d1$, $d2$);

den =

1 6 5

Command Line Editing

The arrow and control keys allow the user to recall edit and reuse the command already entered. For example, suppose that you have typed 'root(p)' instead of 'roots(p)' you will get the error message as shown below.

>> root(p)

??? Undefined function or variable 'root'.

Now press the up arrow ↑ on the keyboard. The previous command line is redisplayed. Press the left arrow ← on the keyboard to move the cursor left and insert s after root. Press the enter key to execute the command.

Suppressing the display on the display device by placing a semicolon ';' after the command as in >>roots(p); MATLAB will perform the computation but does not display the output. This is particularly helpful when the output is large enough to fill the entire screen.

Long Command Lines

If a statement is long enough not to fit on a single line on the display screen, use ellipses (three periods ...) and then press return key to indicate the statement is continued in the next line. For instance

```
x = 1 + 2 + 3 + 4 + 5 + 6 ...
        + 7 + 8 + 9 + 10
```

MARLAB Workspace

The MATLAB workspace contains variables that can be manipulated from the MATLAB command line. Using >> who and >> whos command one can see the variables present in the workspace. The who command gives a short list whereas whos command gives size and data type of the variables. Entering the command >> clear will remove all the variables from the workspace.

Loading and Saving the Workspace

The command

>>save('file_1','x','-ascii')

all single quotes

will save the data *x* with a filename file_1.ascii.

The file_1 may be reloaded into workspace by the command

>> load ('file_1')

M-Files

Files that contain MATLAB language code are called M files. M files can be functions that accept arguments and produce output or they can be scripts that execute a series of MATLAB statements. For MATLAB to recognize a file as an M-file, its name must have .m extension.

The M-file is created using a text editor, then it is used like any other MATLAB functions or command .

There are two kinds of M-files : Script M-files and Function M-files

Scripts

Scripts are simplest of the two M-file—they have no input or output arguments. They are useful for automatically executing a series of MATLAB commands, that are normally performed repeatedly from command line. Scripts operate on existing data in the workspace, or they can create new data on which to operate. Any variables that scripts create remain in the workspace after the scripts finishes, so one can use them for further computations.

The structure of a script M-file is shown below:

Comment lines	% An M-file script to calculate the step response of a second order system given by % $Y(s)/R(s) = \omega_n^2/(s^2 + 2\delta\, \omega_n\, s + \omega_n^2)$ with variable damping ratio
Clear workdpace	clear all;
Closes all the open figure windows, including hidden ones.	close all hidden
Comment lines	% $d = 2\, \delta\, \omega_n$ d=[0.2 1] ; % delta = 0.2, 0.5 num=[1]; % $\omega_n = 1$ t=[0:0.1:12];
Computation	for i=1:2 d1=d(i) den=[1 d1 1]; [y]=step(num,den,t); Ys(:,i)=y; End

Graphical output plot(t, Ys(:,1),'–',t,Ys(:,2),'–')
commands xlabel('Time(Sec)'),ylabel('y(t)')

Save the M-file in folder 'works' as sample1.*m* (do not use the file name 'step.*m*' used by MATLAB). This file is now an M-file. Typing >> sample 1 from MATLAB command line will; execute the statements in the script. When execution is complete the variables Y*s*, *d*, *d*1,den, I, num,t and *y* will reain in the workspace.

Function M-Files

Function M-files can accept input arguments and return output arguments and the internal variables are local to the functions by default. A basic function M-file consists of :

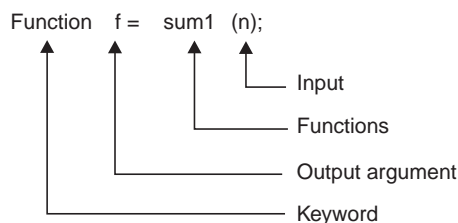
(*a*) a *functions* definition line

(*b*) A *H1* line

(*c*) *Help* text

(*d*) The *functions* body

(*e*) *Comments*

The *functions* definition line informs MATLAB command that the M-file contains a function, and specifies the argument calling sequence of the functions. The *functions* definition line for a function with one input argument and one output argument is shown below:

If the function has more than one output arguments, these are enclosed in square brackets and if more than one input arguments are present, these are put in the parenthesis as shown below :

function [*f*, *n*] = sum1 (*n*, *x*);

In absence of any output argument, the output may be left blank or empty square brackets.

The H1 line is so named because it is the first help text line, which is a comment line and is displayed by typing the function name sum1at the MATLAB command prompt.

One can create online help for the M-files created by the user by entering texts immediately following the H1 line.

The function body contains the MATLAB code for computation of the function and assigning the output arguments.

The comment lines begin with percentage sign (%) and it can appear anywhere in the M-file. A simple function example is given below.

```
function f=sum1(n);
% sum of squares
% It computes the sum of squares of natural numbers up to 10
n=input('enter a value of n in the range 1 to 10, n= ')
```

```
if n>=1 & n<=10;
S=0;
for i=1:n;
S=S+power(i,2);
f=S;
end
else error(‘ n should be > 0 and less than equal to 10. Try again ’)
end
```

There is a vast collection of computational algorithms ranging from elementary functions like *sum, sine, cosine* and complex arithmetic to more sophisticated functions like *matrix inverse*, matrix *eigenvalues*, *Bessel functions* and fast Fourier Transforms. Some functions used in the script files in the text are given in the table below.

By entering >> help ‘function_name’ in the command window will display the formats of using the function. For example >> help tf on the command window will display the command format, along with various options, of using the MATLAB function ‘tf ’ for creating transfer functions.

Table: List of some functions used in the script files in the book

abs	*bode*	*det*	*feedback*	*initial*	*log2*	*margin*	*ode45*	*reshape*	*syms*	*subplot*
acker	*c2d*	*disp*	*gaussmf*	*int*	*loglog*	*max*	*place*	*residue*	*series*	*tf*
and	*clear*	*dlqr*	*hold*	*inv*	*lqr*	*nichols*	*plot*	*rlocus*	*simplify*	*trapmf*
atan	*close*	*eig*	*ilaplace*	*laplace*	*lsim*	*norm*	*poly*	*rltool*	*ss*	*trimf*
atan2	*conv*	*exp*	*imag*	*length*	*ltiview*	*nyquist*	*power*	*roots*	*ssdata*	*xlabel*
axis	*cos*	*expm*	*impulse*	*log10*	*lyap*	*ode23*	*real*	*round*	*step*	*zpk*

Bibliography

[1] Zadeh, L.A., *The Concept of a Linguistic Variable and Its Application to Approximate Reasoning* I, II, III, Information Sciences 8, pp. 199-251, pp. 301-357; 9 pp. 45-80.

[2] Zadeh L.A., *Fuzzy Sets*, *Information and Control,* Vol. 8, pp. 338-353, 1965.

[3] Bezdek, J. *Editorial; Fuzzy Models—What are They and Why?,* IEEE Trans Fuzzy Sys. Vol. 1, pp. 1-5. 1993.

[4] Blakelock, J.H., *Automatic Control of Aircraft and Missiles*, John Wiley & Sons, New York, 1965.

[5] Eykhoff, P.,*System Identification—Parameter and State Estimation*, John Wiley & Sons, London, 1977.

[6] Kolata, G., *Math Problems Long Baffling, Slowly Yields*, New York Times, March 12, 1990 p, C1.

[7] Astrom, K.J. and B. Wittenmark,*Adaptive Control*, Reading, Addison-Wesley Publishing Company, Massachusetts, 1989.

[8] Francis, B.A., *A Course in H Control Theory,* Lecture Notes in Control and Information Sciences, No. 88, Springer-Verlag, Berlin, 1987.

[9] Kalman, R.E.,*On the General Theory of Control Systems,* Proc. IFAC, Moscow Congress, Vol. I, pp. 481-492, Butterworth Inc, Washington D.C., 1960.

[10] Gardner M.J. and J. L. Barnes, *Transients in Linear Systems,* John Wiley & Sons Inc., New York, 1942.

[11] Van der Pol, B. and H. Bremmer, *Operational Calculus Based on Two-Sided Laplace Integral,* Cambridge University Press, Cambridge, England, 1950.

[12] Mason, S.J., *Feedback Theory: Some Properties of Signal Flow Graphs*, Proc. IRE, Vol. 41, No. 9, pp. 1144-1156, Sept. 1953.

[13] Mason, S.J., *Feedback Theory: Further Properties of Signal Flow Graphs*, Proc. IRE, Vol. 44, No. 7, pp. 920-926, July, 1956.

[14] Gantmacher, F.R., *Theory of Matrices, Vol I and Vol II*, Chelsea Publication Co. Inc., New York, 1959.

[15] Bode, H.W., *Relation Between Attenuation and Phase in Feedback Amplifier Design*, Bell System Tech. J. pp. 421 to 454, July, 1940.

[16] Routh, E.J., *Dynamics of a System with Rigid Bodies,* Mcmillan, New York, 1892.

[17] Hurwitz, A., *On the Condition Under which an Equation has only Roots with Negative Real Parts*, Mathematische annalen 46 1895 pp. 273-384. vide *Selected Papers on Mathematical Trends in Control Theory*, pp. 70-82, Dover, New York 1964.

[18] Liapunov, A.M., *On the General Problem of Stability of Motion,* Ph.D. Thesis.

[19] Kalman, R.E. and J.E. Bertram, *Control System Analysis and Design via the Second Method of Liapunov*, Parts I and II, Trans ASME, Vol. 82, Series D, pp. 371-400, 1960.

[20] Schultz, D.G. and J.L. Melsa, *State Functions and Linear Control Systems*, New York, McGraw-Hill Book Company, 1967.

[21] Schultz, D.G., *The Generations of Liapunov Functions in C.T. Leondes* (ed.) *Advances in Control System Vol 2,* Academic Press Inc, New York, 1965.

[22] Mandal, A.K., *Lyapunov Function for a Time Varying Nonlinear System,* IEEE Trans Auto Control pp. 570-571, Aug. 1972.

[23] Mandal, A.K., Choudhury A.K. and D. Roy Choudhury, *Construction of Lyapunov Function Staring from a Initiating Function*, Proc IEE-IERE, India, Vol. 12, No. 1. pp. 3-14, Jan-Feb., 1974.

[24] Nyquist, H., *Regeneration Theory,* Bell System Technical Journal, January, pp. 126-147, 1932.

[25] Yeung, K.S. and *A Reformulation of Nyquists Criterion,* IEEE Trans Educ. Vol E-28, pp. 58-60, Feb., 1985.

[26] Yeung, K.S., H.M. Lai, *A Reformulation of Nyquist Criterion for Discrete Systems,* IEEE Trans Educ. Vol. E-31 No. 1, Feb., 1988.

[27] James, H.M., N.B. Nichols and R.S. Philips, *Theory of Servomechanisms*, McGraw-Hill Book Company, Inc., New York, 1947.

[28] Evans, W.R., *Graphical Analysis of Control Systems,* Transactions of AIEE, 67, pp. 547-551, 1948.

[29] Ziegler, J.G. and N.B. Nichols, *Optimum Settings for Automatic Controllers,* ASME Trans 64, pp. 759-768, 1942.

[30] Cohen, M.L., *A Set of Stability Constraint on the Denominator of a Sample Data Filter,* IEEE Trans Automatic Control, Vol. AC-11, pp. 327-328, April 1966.

[31] Jury, E.I. and J. Blanchard, *A Stability Test for Linear Discrete-time Systems in Table Form*, Proc. IRE, Vol. 49, pp. 1947-1948, 1961.

[32] Raible, R.H., *A Simplification of Jury's Tabular Form,* IEEE Trans Automatic Control, Vol. Ac-19, pp. 248-250, June 1974.

[33] Pontryagin, L.S., V.G. Boltianskii, R.V. Gamkrelidze and E.F. Mischenko, *The Mathematical Theory of Optimal Processes*, Interscience Publishers, Inc., New York, 1962.

[34] Bellman, R., *On the Determination of Optimal Trajectories Via Dynamic Programming* G. Lietman etd Optimization Techniques, Academic Press, New York, 1962.

[35] Elsgole, L.E., *Calculus of Variations,* Addison Wesley Publishing Company, Inc., Reading, Mass., 1962.

[36] Gelfand, I.M. and S.V. Fomin, *Calculus of Variations,* Prentice Hall, Inc., Englewood Cliffs, N.J.; 1963.

[37] Kirk, D.E., *Optimal Control Theory—An Introduction,* Prentice Hall, Inc., Englewood Cliffs, N.J., 1970.

[38] Athans, M., and P.L. Falb, *Optimal Control: An Introduction to the Theory and Its Applications*, McGraw-Hill Book Company, New York, 1966.

[39] Sage, A.P., *Optimum Systems Control*, Englewood Cliffs, N.J., Prentice-Hall, Inc., 1968.

[40] Sugeno, M., *Fuzzy Measures and Fuzzy Integrals: A Survey,* Fuzzy, Automata and Decision Processes etd : M. Gupta, M.G.N. Saridis and B.R. Gaines, North Holland, New York, pp.89-102, 1977.

[41] Yager, R.R., *On a General Class of Fuzzy Connectives*, Fuzzy Sets and Systems, 4, No. 3, pp. 235-242.

[42] Dombi, J., *A General Class of Fuzzy Operators, the De Morgan Class of Fuzzy Operators and Fuzziness Measures Induced by Fuzzy Operators*, Fuzzy Sets and Systems, 8, No. 2, pp. 149-163.

[43] Dubois, D. and H. Prade, *Fuzzy Sets and Systems Theory and Applications*, Academic Press, Inc., Orlando, Florida.

[44] Mamdani, E.H., *Application of Fuzzy Algorithms for Simple Dynamic Plant*, Proc. IEE 121, 12, 1974, pp. 1585-1588.

[45] Larsen, P.N., *Industrial Application of Fuzzy Logic Control,* Int. J. Man Mach Studies, 12, 1, pp. 3-10, 1980.

[46] Mamdani, E.H. and S. Assilian, *An Experiment with Linguistic Synthesis of a Fuzzy Logic Controller*, Int. J. Man Mach Studies, 7, 1, pp. 1-13, 1975.

[47] Holmblad, L.P. and J.J. Ostergaard, *Control of Cement Kiln by Fuzzy Logic* in M.M. Gupta and E. Sanchez etd, Approximate Reasoning in Decision Analysis pp. 389-400, Amsterdam, North Holland, 1982.

[48] Sugeno, M. and G.T. Kang, *Structure Identification of Fuzzy Model*, Fuzzy Sets and System, 28, No. 1, pp. 15-33, 1988.

[49] Tagaki, T. and M. Sugeno, *Fuzzy Identification of System and its Application to Modeling and Control*, IEEE Trans on System, Man, and Cybernetics, 15, 1, pp. 116-132, 1985.

[50] Filev, D.P. and Yager P.R., *A Generalized Defuzzification Method with BAD Distribution*, Int. J. Intell. System, pp. 687-697, Vol. 6, No. 7, 1991.

[51] Passino, K.M., and S. Yurkovich, *Fuzzy Control*, Addison-Wesley, California, 1998.

[52] Togai, M. and P.P. Wang, *Analysis of a Fuzzy Dynamic System and Synthesis of Its Controller*", Int. J. Man Machine Studies, Vol. 22, pp. 355-363, 1985.

[53] Procyk, T.J. and E.H. Mamdani, *A Linguistic Self-organising Process Controller,* Automatica, pp. 15-30, Vol. 15, No. 1, 1979.

[54] Scharf, E.M. and N.J. Mandic, *The Application of a Fuzzy Controller to the Control of a Multi-degree-freedom Robot Arm* in M. Sugeno (etd). Industrial Application of Fuzzy Control, pp. 41-62, Amsterdam.

[55] Lee, C.C., *Fuzzy Logic in Control Systems*: *Fuzzy Controllers—Part-I, Part-II*, IEEE Trans on System, Man and Cybernetics, Vol. 20, pp. 404-435.

[56] Kailath, T., *Linear Systems,* Prentice Hall, pp. 209-211, 1980.

[57] *MATLAB ver 6.5 and Simulink ver 5.0*, Mathworks Inc., 24 Prime Park Way, Natick, MA, 01760-1520, USA.

[58] Truxal, J.G., *Automatic Feedback Control*, *System Synthesis*, McGraw-Hill Book Company, New York, 1955.

[59] Gibson, J.E., *Nonlinear Automatic Control,* McGraw-Hill Book Company, Inc., New York, 1963.

[60] Thaler, G.J., and R.G. Brown, *Analysis and Design of Feedback Control Systems,* McGraw-Hill Book Company, Inc., New York, 1960.

[61] Minorsky, N., *Theory of Nonlinear Control Systems*, McGraw-Hill Book Company, New York, 1969.

[62] Kokenburger, R.J., *A Frequency Response Method for Analyzing and Synthesizing Contactor Servomechanisms*, Trans AIEE, Vol. 69, Part I, pp. 270-284, 1950.

[63] Tustin, A., *The Effect of Backlash and Speed-Development Friction on the Stability of Closed-Cycle Control System,* J. Inst. Elec. Engrs. (London) Vol. Part IIA, No. 1, May, 1947.

[64] Andromow, A.A. and C.E. Chaikin, *Theory of Oscillations,* Princeton University Press, Princeton, N.J., 1949.

[65] Pell, W.H., *Graphical Solution of Single-degree-of-freedom Vibration Problem with Arbitrary Damping and Restoring Forces* J. Appl. Mech, Vol. 24, pp. 311-312, 1957.

[66] Jacobson, L.S., *On a General Method of Solution of a Second Order Ordinary Differential Equation by Phase Plane Displacements,* J. Appl. Mech, Vol. 19, p. 543-553,1952.

[67] Cunningham, W.J., *Introduction to Nonlinear Analysis,* McGraw-Hill Book Company, Inc., New York, pp. 88-106, 1958.

[68] Franklin, G.F., J.D. Powell and Abbas Emami-Naeini, *Feedback Control of Dynamics Systems*, Reading, Massachusetts, Addison-Wesley Publishing Company, 1986.

[69] Hwang, S.Y., *On Optimization of Cascade Fixed-Point Digital Filters,* IEEE Trans Circuits & Syst. Vol. CAS-21, pp. 163-166, Jan. 1974.

[70] Steiglitz K., B. Liu, *An Improved Algorithm for Ordering Poles and Zeros of Fixed Point Recursive Digital Filters*" IEEE Trans Accous. Speech Signal Process, Vol. Assp-24, pp. 341-343, Aug, 1976.

[71] Ross, T.J., *Fuzzy Logic with Engineering Application,* McGraw-Hill, Inc., New York, 1995.

[72] Li-Xin Wang, *A Course in Fuzzy Systems and Control*, Prentice Hall Inc., NJ, 1997.

[73] Driankov, D., H. Hellendoorn, and M. Reinfrank, *An Introduction to Fuzzy Control*, Springer-Verlag, Berlin, 1993.

[74] Klir, G.J. and B. Yuan, *Fuzzy Sets and Fuzzy Logic—Theory and Application*, Prentice-Hall Inc., New Jersey, 1995.

[75] Graham, D. and R.C. Lathrop, *Synthesis of Optimum Response Criteria and Standard Forms*, AIEE Trans Part II, pp. 273-288, 1953.

[76] Lin, C.T. and C.S.G. Lee, *A Neuro-Fuzzy Synergism to Intelligent Systems*, Prentice-Hall Inc., New Jersey.

[77] Houpis, C.H. and G.B. Lamout, *Digital Control System—Theory, Hardware, Software*, McGraw-Hill Book Company, Inc., New York, 1985.

[78] Phillips, C. and H.T. Nagle, *Digital Control System Analysis & Design,* 2nd ed., Prentice Hall, New Jersey, 1990.

[79] Kuo, B.C., *Digital Control Systems,* 2nd Edition, Oxford, New York, 1992.

[80] Franklin, G.F., J.G. Powell, and M.L. Workman, *Digital Control of Dynamic Systems*, 3rd Edition, Pearson Education, Inc., Reading Mass., 2000.

[81] Leigh, J.R., *Applied Digital Control*, Prentice-Hall, Englewood Cliffs, New Jersey, 1985.

[82] Katz, P., *Digital Control Using Microprocessors*, New Reinhold Company, Englewood Cliffs, 1979.

[83] Kuo, B.C., *Automatic Control Systems*, 6th Edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1991.

[84] Dorf, R.C. and Robert H. Bishop, *Modern Control Systems*, 8th Ed., Addison-Wesley Longman Inc., Harlow, England, 1998.

[85] Chen, C.T., *Control System Design*, Saunders College Publishing, Orlando, Florida, 1993.

[86] Ogata, K., *Modern Control Engineering*, 2nd Edition, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[87] Narendra, K.S. and A.M. Annaswamy, *Stable Adaptive Systems*, Prentice-Hall, Englewood Cliffs, New Jersey, 1989.

[88] Landau, Y.D., *Adaptive Control*: *The Model Reference Approach*, Marcel Dekker, New York, 1979.

[89] DeRusso, P.M., R.J. Roy and C.M. Close, *State Variables for Engineers*, John Wiley & Sons, New York, 1965.

[90] Anderson, B.D.O., and J.B. Moore, *Optimal Control*: *Linear Quadratic Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1990.

[91] Wolovich, W.A., *Linear Multivariable Systems*, Springer-Verlag, New York, 1974.

[92] Isermann, R., *Digital Control Systems*, Vol. II, 2nd Edition, Springer-Verlag, Berlin, 1991.

[93] D'Azzo, J.J. and C.H. Houpis, *Linear Control System Analysis and Design*, 3rd Edition, McGraw-Hill Book Company, New York, 1988.

[94] Ahrendt, W.R. and C.J., Savant, Jr., *Servomechanism Practice*, 2nd Edition, McGraw-Hill Book Company, New York, 1960.

Index

C

D

E

Index

F

L

M

Q

R

S

T